



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Everton Gomedes

Arquiteturas de Redes Neurais Artificiais para Aprendizado Adaptativo em Sistemas de E-learning

Campinas

2020

Everton Gomedede

Arquiteturas de Redes Neurais Artificiais para Aprendizado Adaptativo em Sistemas de E-learning

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica, na Área de Telecomunicações e Telemática.

Orientador: Prof. Dr. Leonardo de Souza Mendes

Este trabalho corresponde à versão final da tese defendida pelo aluno Everton Gomedede, e orientada pelo Prof. Dr. Leonardo de Souza Mendes.

Campinas

2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

G585a Gomedede, Everton, 1979-
Arquiteturas de redes neurais artificiais para aprendizado adaptativo em sistemas de e-learning / Everton Gomedede. – Campinas, SP : [s.n.], 2020.

Orientador: Leonardo de Souza Mendes.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Redes neurais (Computação). 2. Inteligência artificial. 3. Inteligência artificial - Aplicações educacionais. 4. Inteligência computacional. 5. Sistemas de reconhecimento de padrões. I. Mendes, Leonardo de Souza, 1961-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Architecture of artificial neural networks for adaptive learning in e-learning systems

Palavras-chave em inglês:

Neural networks (Computing)

Artificial intelligence

Artificial intelligence - Educational applications

Computational intelligence

Pattern recognition systems

Área de concentração: Telecomunicações e Telemática

Titulação: Doutor em Engenharia Elétrica

Banca examinadora:

Leonardo de Souza Mendes [Orientador]

Dalton Soares Arantes

André Macedo Panhan

Rodolfo Miranda de Barros

Paulo Cardieri

Data de defesa: 27-07-2020

Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0003-4839-9017>

- Currículo Lattes do autor: <http://lattes.cnpq.br/3874575351179371>

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato(a): Everton Gomedes RA: 180565

Data da defesa: 27 de julho de 2020

Titulo da Tese: "Arquiteturas de Redes Neurais Artificiais para Aprendizado Adaptativo em Sistemas de E-learning"

Prof. Dr. Leonardo de Souza Mendes (Presidente)

Prof. Dr. Dalton Soares Arantes

Prof. Dr. André Macedo Panhan

Prof. Dr. Rodolfo Miranda de Barros

Prof. Dr. Paulo Cardieri

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

*Dedico esta tese à Karla e Mariana;
também a memória do meu pai e a força da minha mãe.*

Agradecimentos

Agradeço aos colegas de pesquisa Gabriel e Fernando pelas viagens e conversas durante o período dos créditos. Agradeço aos amigos que fiz durante o trabalho e, para evitar o risco de esquecer alguém, não vou citar uma lista com o nome de todos. Agradeço ao pessoal da secretaria da FEEC que sempre me ajudou com as questões administrativas. Agradeço também ao pessoal da DAC e CPG pelo suporte para resolver questões relacionadas ao aproveitamento de disciplinas e demais conteúdos. Um agradecimento especial para meu amigo Rogério que sempre me ajudou, desde o começo desta jornada, com a complicada logística envolvendo o trabalho como cientista da computação, professor e consultor com o tempo dedicado a esta pesquisa. Agradeço também aos professores e orientadores Leonardo e Rodolfo pela ajuda, orientação, paciência e, principalmente, pela visão de mundo compartilhada que permitiu com que um aluno pudesse aprender as técnicas de pesquisa em ciência da computação em um sentido mais amplo e profundo. Por fim, agradeço a Unicamp por proporcionar uma educação pública de qualidade.

“Keep moving forward.”
(Meeting the Robinsons, Walt Disney, 2007)

Resumo

O problema de Aprendizado Adaptativo pode ser descrito como um conjunto de preferências \mathbf{P} , do aluno A , que implica na interação com o conjunto de objetos de aprendizagem \mathbf{O} , contidos no sistema de *e-learning*; portanto, $\mathbf{P} \rightarrow \mathbf{O}$. Os dados desta interação permitem gerar modelos que expliquem o comportamento de tal aluno e, adicionalmente, prever seu comportamento. As redes neurais artificiais desempenham um papel importante para a modelagem de soluções, em diferentes tipos de problemas nos mais variados contextos. Para investigar o problema de Aprendizado Adaptativo, duas arquiteturas de redes neurais artificiais foram testadas; uma para classificação baseada na arquitetura *Multi Layer Perceptron* e outra para recomendação baseada na arquitetura *Deep Auto Encoder*. Como resultado, obtiveram-se duas estratégias, sendo a primeira relacionada com a classificação de um aluno no modelo de estilos de aprendizagem de Felder-Silverman e a segunda uma lista de objetos de aprendizagem que possam ser recomendados de acordo com as preferências de aprendizagem. Ambas as arquiteturas se mostraram eficazes permitindo que conteúdos e/ou formatos possam ser entregues, nos sistemas adaptativos de *e-learning*, de maneira adequada às preferências de seus usuários. Dentre as implicações práticas, pode-se destacar a possibilidade de melhora na experiência de aprendizagem do aluno devido à adaptação de conteúdos e/ou formatos.

Palavras-chaves: Arquiteturas de Redes Neurais Artificiais; Sistemas Adaptativos de *E-learning*; *Multi Layer Perceptron*; *Deep Auto Encoder*.

Abstract

The Adaptive Learning problem can be described as a set of objects \mathbf{P} , from student A , which implies interaction with the set of learning objects \mathbf{O} , in an e-learning system; therefore, $\mathbf{P} \rightarrow \mathbf{O}$. The data of this interaction allows generate models that explain the student's behavior and, in addition, predict his behavior. The artificial neural networks have played an important role in modeling solutions, in different types of problems in several contexts. To investigate the Adaptive Learning problem, two artificial neural network architectures were tested; a specific classification in the Multi Layer Perceptron architecture and another recommended recommendation based on Deep Auto Encoder architecture. As a result, two strategies were obtain, the first one related to the classification of student in the Felder-Silverman Learning Style Model and the second one related to a list of recommended objects suitable to student's learning preferences. Both architectures have proven to be effective, allowing content and/or formats to be delivered, in adaptive e-learning systems, in a manner appropriate to the preferences of its users. Among the practical implications, the possibility of improving the student's learning experience due to the adaptation of contents and/or formats.

Keywords: Artificial Neural Network Architectures; Adaptive E-learning Systems; Multi Layer Perceptron; Deep Auto Encoder.

Lista de ilustrações

Figura 2.1 – Processo para classificação automática de estilos de aprendizagem. Os pontos estudados nesta tese foram os A, B e C. A linha tracejada mostra a detecção via questionário (GOMEDE; BARROS; MENDES, 2020).	45
Figura 2.2 – Resumo da evolução da computação aplicada ao contexto educacional. Pode-se notar que, a partir dos anos 60, as primeira soluções começaram a ser propostas com o objeto de ser um ferramental a disposição dos interessados no ambiente.	54
Figura 3.1 – Exemplo de uma rede neural <i>perceptron</i> . Normalmente, a faixa de amplitude normalizada da entrada/saída de um neurônio é escrita como o intervalo unitário fechado $[0, 1]$ ou, alternativamente, $[-1, 1]$ (HEIDARI et al., 2020).	55
Figura 3.2 – Exemplo de uma rede neural de múltiplas camadas (<i>multi layer perceptron</i>). Normalmente, a faixa de amplitude normalizada da entrada/saída de um neurônio é escrita como o intervalo unitário fechado $[0, 1]$ ou, alternativamente, $[-1, 1]$. Além disso, a quantidade de neurônios e camadas são hiperparâmetros da rede que devem ser otimizados para o problema em questão (HEIDARI et al., 2020).	57
Figura 3.3 – Um <i>Deep Auto Encoder</i> (DAE) é uma categoria de algoritmos que são formados por duas partes, um <i>encoder</i> e um <i>decoder</i> . O <i>encoder</i> tem como função comprimir a informação da entrada em um espaço-latente diferente. Pode ser representado como um função de $x, f(x) = h$	59
Figura 3.4 – A retropropagação é um algoritmo amplamente utilizado no treinamento de redes neurais artificiais <i>feedforward</i> para aprendizado supervisionado ou semi-supervisionado.	60
Figura 3.5 – Duas estratégias para a divisão do conjunto de dados. A divisão em treinamento e testes e a divisão em treinamento, validação e testes. A estratégia mais comum é a de treinamento e testes (em destaque) (RASCHKA, 2018).	65
Figura 3.6 – O método <i>Cross-Validation</i> divide o conjunto de dados em k amostras (<i>folds</i>). Cada uma delas contém um número de observações. A cada iteração $k - 1$ amostras são selecionadas para o treinamento e a restante para o teste (RASCHKA, 2018).	67
Figura 3.7 – O método <i>Leave-one-out Cross-Validation</i> (LOOCV) é um caso extremo do método <i>k-fold cross validation</i> . Uma amostra é usada para teste e as demais para treinamento (RASCHKA, 2018).	68

Figura 3.8 – O método de <i>Bootstrap</i> cria várias amostragens que sejam o mais próximas possíveis do conjunto de dados original. Estas amostras são utilizadas para criação de modelos diferente. Estes modelos são comparados para verificar o resultado de cada um deles. Este tipo de método é conhecido como <i>ensembler</i> (RASCHKA, 2018).	69
Figura 3.9 – Primeira iteração	70
Figura 3.10–Espaço de cobertura	70
Figura 3.11–Divisão do conjunto de dados em dois subconjuntos que cobrem aproximadamente a mesma região do espaço fatorial definido pelo conjunto de dados original (RASCHKA, 2018).	70
Figura 3.12–Exemplo de taxonomia utilizada para objetos de aprendizado em um LMS.	72
Figura 3.13–Repositórios de Objetos de Aprendizagem são como depósitos virtuais onde ficam armazenados os materiais com fins educacionais para um sistema de <i>e-learning</i>	74
Figura 4.1 – A variabilidade dos comportamentos dos alunos para os objetos de aprendizagem. Na primeira figura, os dados não estão normalizados. Na segunda figura, os dados foram normalizados de acordo com a Equação 4.1 para aumentar a acurácia do modelo.	79
Figura 4.2 – A distribuição da preferência dos alunos entre $[-11 : 11]$ para cada dimensão. Não são determinísticas, e sim uma tendência (FELDER; SILVERMAN et al., 1988).	80
Figura 4.3 – O processo utilizado para a rotulação dos dados. Os dados foram obtidos pela adaptação do formulário de Felder-Silverman. (FELDER; SILVERMAN et al., 1988).	81
Figura 4.4 – Comportamento da função de perda RMSE do treinamento da rede MLP.	87
Figura 4.5 – O fenômeno de cauda longa para a quantidade de tempo gasto com os objetos de aprendizagem. A grande parte dos alunos gasta entre 0 e 2 horas em cada vídeo. Esta quantidade de alunos tem um decaimento logarítmico conforme o tempo aumenta o que representa que muitos alunos gastam pouco tempo nos objetos de aprendizagem.	89
Figura 4.6 – Alunos	90
Figura 4.7 – Objetos	90
Figura 4.8 – O fenômeno de cauda longa também acontece para os demais elementos do conjunto de dados. Por exemplo, pode-se notar que a frequência de alunos (a) tem um comportamento logarítmico, sendo que poucos alunos acessam frequentemente os objetos de aprendizagem. Da mesma forma ocorre com os objetos de aprendizagem (b); poucos são acessados frequentemente.	90

Figura 4.9 – A matriz esparsa resultante da transposição do conjunto de dados. Esta matriz indica que poucos alunos acessam poucos objetos de aprendizagem. Desta forma, têm-se um grande numero de objetos e de alunos que não foram beneficiados dos conteúdos que estão escondidos.	94
Figura 4.10–Alunos	95
Figura 4.11–Objetos	95
Figura 4.12–(a) O fenômeno de cauda longa representado pela distribuição da frequência de acesso dos alunos pela quantidade de horas. (b) Os alunos que acessam objetos por mais de 5 horas são considerados como <i>outliers</i> . . .	95
Figura 4.13–Análise de agrupamento utilizando a técnica PCA que mostra a concentração da matriz em alguns alunos e objetos de aprendizagem. Os componentes são compostos das características <i>student_id</i> , <i>object_id</i> e <i>hours</i> . Foram normalizados para permitir a análise. Pode-se notar que o agrupamento dos <i>PC1</i> , <i>PC2</i> e <i>PC3</i> indicam que poucos alunos acessam poucos objetos com frequência.	96
Figura 4.14–Histórico das interações aluno/objeto coletado durante os anos de 2018 e 2019. Pode-se notar que o maior volume encontra-se durante os meses de março a novembro e os horários de acesso entre <i>08h00min</i> e <i>19h00min</i>	97
Figura 5.1 – A distribuição das observações do conjunto de dados de cada dimensão. O % das preferências de cada aluno é representada como 48% sequenciais, 52% globais, 50% visuais e 50% verbais, 41% intuitivos e 59% reflexivos, 51% ativos e 49% reflexivos.	106
Figura 5.2 – A análise de <i>boxplot</i> do conjunto de dados (descritores e metas de contagem e tempo). Há uma limitação para o descritor de tempo em 120 e a contagem em 10 vezes.	106
Figura 5.3 – Distribuição da frequência do estilo de aprendizagem de cada dimensão. Tal distribuição aparentemente não mostra nenhum padrão.	107
Figura 5.4 – <i>Clusters</i> naturais no conjunto de dados usando médias de <i>k</i> (com $k = 2, 3, 4, 5$). Nenhum <i>cluster</i> natural foi identificado e, portanto, as classes não são lineares separadas (HEIDARI et al., 2020).	108
Figura 5.5 – Análise de Componentes Principais (PCA) de cada dimensão, processamento, percepção, entrada e entendimento, respectivamente. Pode-se notar que nenhuma das características analisadas possui maior relevância relativa.	109
Figura 5.6 – O modelo construído com 2 camadas ocultas, 26 neurônios de entrada e 4 neurônios de saída. Os pesos representados em cada sinapse (conexão) representam o conhecimento induzido pelo treinamento da rede.	110
Figura 5.7 – Modelo	111
Figura 5.8 – Treinamento	111

Figura 5.9 – A análise ROC de cada dimensão da matriz de confusão que ilustra a capacidade de diagnóstico do sistema classificador binário, pois seu limiar de discriminação é variado. (a) Representa a matriz de confusão individual e (b) representa a curva ROC individual.	111
Figura 5.10–Mean Average Precision	112
Figura 5.11–Métricas de Personalização	112
Figura 5.12–Resultado com as métricas do modelo de popularidade. Este modelo não fornece personalização mas é utilizado como base para a comparação dos demais modelos gerados. Pode-se notar que a métrica <i>personalization</i> apresentou o maior valor relativo (a) e (b).	112
Figura 5.13–Mean Average Precision	115
Figura 5.14–Métricas de Personalização	115
Figura 5.15–Resultado com as métricas do <i>Collaborative Denoising Auto-Encoders</i> . Este modelo fornece uma boa personalização e é utilizado para a comparação dos demais modelos gerados. Pode-se notar que a métrica <i>personalization</i> apresentou o maior valor relativo (a) e (b).	115
Figura 5.16–Modelo	115
Figura 5.17–Treinamento	115
Figura 5.18–Modelo treinado utilizando o conjunto de dados particionados. Este modelo possui as camadas de entrada e saída de dados (ambos do mesmo tamanho) e as camadas ocultas com as matrizes fatoradas. Histórico do treinamento da rede. Pode-se notar o mesmo comportamento do treinamento da rede MLP. A taxa de erro é mais alta no início, uma rápida redução e a convergência da rede.	115
Figura 5.19–Mean Average Precision	117
Figura 5.20–Métricas de Personalização	117
Figura 5.21–Resultado com as métricas do <i>Deep Auto Encoder for Collaborative Filtering</i> . Este modelo fornece uma boa personalização e é utilizado para a comparação dos demais modelos gerados. Pode-se notar que a métrica <i>personalization</i> apresentou o maior valor relativo (a) e (b).	117
Figura 5.22–Modelo	117
Figura 5.23–Treinamento	117
Figura 5.24–(a) Modelo treinado utilizando o conjunto de dados particionados. Este modelo possui as camadas de entrada e saída de dados (ambos do mesmo tamanho) e as camadas ocultas com as matrizes fatoradas. (b) Histórico do treinamento da rede. Pode-se notar o mesmo comportamento do treinamento da rede MLP. A taxa de erro é mais alta no início, uma rápida redução e a convergência da rede.	117
Figura 5.25–Mean Average Precision	119

Figura 5.26–Métricas de Personalização	119
Figura 5.27–Resultado com as métricas do <i>Deep Auto Encoder With Content Information</i> . Este modelo fornece uma boa personalização e é utilizado para a comparação dos demais modelos gerados. Pode-se notar que a métrica <i>personalization</i> apresentou o maior valor relativo (a) e (b).	119
Figura 5.28–Modelo	119
Figura 5.29–Treinamento	119
Figura 5.30–Modelo treinado utilizando o conjunto de dados particionados. (a) Este modelo possui as camadas de entrada e saída de dados (ambos do mesmo tamanho) e as camadas ocultas com as matrizes fatoradas. (b) Histórico do treinamento da rede. Pode-se notar o mesmo comportamento do treinamento da rede MLP. A taxa de erro é mais alta no início, uma rápida redução e a convergência da rede.	119
Figura 5.31–Distribuição	120
Figura 5.32–Frequência	120
Figura 5.33–A distribuição do <i>score</i> gerado apresenta alguns valores como <i>outliers</i> . Estes valores contribuem positivamente para métricas como top@N (onde N é o número de objetos recomendados) para N com valores menores (por exemplo, $N = 10$). Em (a) pode-se notar o comportamento logarítmico de decaimento. Em (b) pode-se notar a frequência dos valores de <i>score</i>	120
Figura 5.34–Os primeiros objetos a serem recomendados apresentam-se como <i>outliers</i> se distanciando dos demais.	121
Figura 5.35–A comparação dos resultados, para cada métrica disponível, dos trabalhos publicados. Pode-se notar que dois trabalhos apresentaram resultados mais expressivos. Entretanto, nenhum deles apresentou um conjunto de métricas mais abrangente.	121
Figura C.1–Taxonomia para as técnicas empregadas em sistemas de recomendação. Dentre cada um dos grupos pode se utilizar algoritmos de redes neurais entre outros.	142
Figura D.1–A visão geral do experimento de classificação. As entradas da rede representam o comportamento do aluno ao interagir com os objetos de aprendizagem. A saída representa os estilos de aprendizagem do FLSM.	143
Figura E.1–A visão geral do experimento de recomendação. A entrada da rede representa o comportamento do aluno ao interagir com os objetos de aprendizagem. A saída esta associada com a previsão do comportamento do mesmo.	144

Figura A.1 – Resultado da aplicação do questionário do modelo de estilos de aprendizagem de Felder-Silverman. Os resultados para cada dimensão indicam a tendência na preferência do aluno. 146

Lista de tabelas

Tabela 2.1 – Comparação entre os trabalhos relacionados com relação as questões técnicas	52
Tabela 2.2 – Comparação entre os trabalhos relacionados com relação as questões educacionais e/ou pedagógicas	53
Tabela 3.1 – Resumo das estratégias para divisão do conjunto de dados em subconjuntos	66
Tabela 3.2 – Resumo com os algoritmos para divisão de dados	71
Tabela 3.3 – Modelo de estilos de aprendizagem de Felder-Silverman	75
Tabela 4.1 – Variáveis que capturam a interação aluno/objeto (BERNARD et al., 2017)	78
Tabela 4.2 – Variáveis Alvo	80
Tabela 4.3 – Matriz de Confusão para um classificador binário	85
Tabela 4.4 – Exemplo dos dados brutos obtidos do ambiente de cursos <i>on-line</i>	88
Tabela 4.5 – Transformada do conjunto de dados na matriz de interação aluno/objeto	88
Tabela 4.6 – Interpretação dos <i>feedback</i>	92
Tabela 4.7 – Resumo do conjunto de dados	93
Tabela 4.8 – Exemplo de uma matriz esparsa. As células com ? representam a ausência de interações aluno/objeto	98
Tabela 4.9 – Exemplo de uma matriz reconstruída por um <i>Deep Auto Encoder</i> . As células destacadas representam o valor previsto da interações aluno/objeto	98
Tabela 4.10–Hiperparâmetros utilizados nos <i>Auto Encoder</i> (ORDWAY-WEST; PARVEEN; HENSLEE, 2018)	99
Tabela 5.1 – Métricas e respectivos resultados para cada dimensão	110
Tabela 5.2 – <i>Collaborative Denoising Auto Encoders</i> - Hiperparâmetros	113
Tabela 5.3 – Lista dos top@10 gerados por <i>Collaborative Denoising Auto Encoders</i> .	116
Tabela 5.4 – <i>Deep Auto Encoder for Collaborative Filtering</i> - Hiperparâmetros	116
Tabela 5.5 – <i>Deep Auto Encoder for Collaborative Filtering</i>	118
Tabela 5.6 – <i>Deep Auto Encoder for Collaborative Filtering With Content Information</i>	118
Tabela 5.7 – <i>Deep Auto Encoder for Collaborative Filtering With Content Information</i>	120
Tabela 5.8 – <i>Input</i> - A matriz esparsa com os dados original das interações do aluno 25 com os objetos de aprendizagem.	122
Tabela 5.9 – <i>Output</i> - Assinatura de Preferência do Aluno 25	122
Tabela A.1–Questionário de avaliação dos estilos de aprendizagem – 1/4	147
Tabela A.2–Questionário de avaliação dos estilos de aprendizagem – 2/4	148
Tabela A.3–Questionário de avaliação dos estilos de aprendizagem – 3/4	149
Tabela A.4–Questionário de avaliação dos estilos de aprendizagem – 4/4	150

Lista de abreviaturas e siglas

SCORM	O <i>Sharable Content Object Reference Model</i> é uma coleção de padrões e especificações para objetos de aprendizagem
TIC	É um conjunto de recursos tecnológicos integrados entre si, que proporcionam, por meio das funções de hardware, software e telecomunicações, a automação e comunicação dos processos de negócios, da pesquisa científica e de ensino e aprendizagem
DEA	<i>Deep Auto Encoders</i> consistem em duas redes profundas idênticas; uma rede para codificação e outra para decodificação. Os <i>Auto Encoders</i> profundos tipicamente têm de 4 a 5 camadas para codificação e decodificação
MOOC	Um Curso Online Aberto e Massivo, <i>Massive Open Online Course</i> (MOOC), é um tipo de curso aberto oferecido por meio de ambientes virtuais de aprendizagem que visam oferecer, para um grande número de alunos, a oportunidade de ampliar seus conhecimentos em um processo de co-produção
LMS	Um Sistema de Gestão da Aprendizagem, <i>Learning Management System</i> (LMS), também chamado de sistema de <i>e-learning</i> disponibiliza uma série de recursos, síncronos e assíncronos, que dão suporte ao processo de aprendizagem, permitindo seu planejamento, implementação e avaliação
RMSE	O desvio quadrático médio quadrático ou o erro médio quadrático é uma medida frequentemente usada para detectar as diferenças entre os valores previstos por um modelo e os valores observados
AE	<i>Auto Encoders</i> são redes neurais treinadas com o objetivo de copiar o seu <i>input</i> para o seu <i>output</i>
ANN	Redes Neurais Artificiais, <i>Artificial Neural Networks</i> (ANN), são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento por meio da experiência
MAE	O <i>Mean Absolute Error</i> é uma medida da diferença entre duas variáveis contínuas

NMAE	O <i>Normalized Mean Absolute Error</i> normaliza o MAE pelo intervalo de valores de classificação disponíveis
NMF	O <i>Non-negative Matrix Factorization</i> é um método de fatoração matricial restrita as matrizes a não-negativas
PCA	O <i>Principal Component Analysis</i> é um procedimento que utiliza a transformação ortogonal para converter um conjunto de observações correlacionadas em um conjunto variáveis linearmente não correlacionadas chamadas de componentes principais
SVD	O <i>Singular Value Decomposition</i> é uma fatoração de uma matriz real ou complexa que generaliza a composição automática de uma matriz quadrada normal para qualquer matriz por meio de uma extensão da decomposição polar
MLP	O <i>Multi Layer Perceptron</i> é uma rede neural semelhante à <i>perceptron</i> , mas com mais de uma camada de neurônios em alimentação direta
tSNE	A <i>t-Distributed Stochastic Neighbor Embedding</i> é uma técnica não linear para redução de dimensionalidade que é particularmente adequada para a visualização de conjuntos de dados de alta dimensão
XML	A <i>Extensible Markup Language</i> é uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais
AICC	O <i>Aviation Industry Computer-Based Training Committee</i> é o primeiro padrão de comunicação para WBT (web-based training), surgiu em 1988 e serviu de base para todos os outros padrões posteriores
RTE	O <i>Run-Time Environment</i> lida com requisitos para iniciar objetos de conteúdo, estabelecendo comunicação entre sistemas de gerenciamento de aprendizagem (LMSs) e objetos de conteúdo compartilháveis
FSLSM	O <i>Felder-Silverman Learning Style Model</i> é o modelo de estilos de aprendizagem criado por Felder e Silverman
HTML	<i>HyperText Markup Language</i> é uma linguagem de marcação utilizada na construção de páginas na Web
MAP	A média da precisão geral, <i>Mean Average Precision</i> (MAP), é a métrica utilizada para indicar o quanto um recomendador é acurado
IP	O <i>Internet Protocol</i> é um rótulo numérico atribuído a cada dispositivo conectado a uma rede de computadores que utiliza o Protocolo de Internet para comunicação

SELU	A <i>Scaled Exponential Linear Unit</i> é uma função de ativação auto-normalizando para rede neural
NDCG	A <i>Normalized Discounted Cumulative Gain</i> é uma medida da qualidade do ranking usada para medir a eficácia dos algoritmos dos mecanismos de pesquisa
ROC	A <i>Receiver Operating Characteristic Curve</i> é uma representação gráfica que ilustra o desempenho de um sistema classificador binário à medida que o seu limiar de discriminação varia
IEEE	O Instituto de Engenheiros Eletricistas e Eletrônicos, <i>Institute of Electrical and Electronic Engineers</i> , é uma organização profissional sem fins lucrativos, fundada nos Estados Unidos
SCO	Os Objetos de Conteúdo Compartilháveis, <i>Shareable Content Objects</i> , representam o menor nível de granularidade que um LSM pode gerenciar
CAM	O Modelo de Agregação de Conteúdo, <i>Content Aggregation Model</i> , especifica como um conteúdo deve ser empacotado para distribuição
SN	O Sequenciamento e a Navegação, <i>Sequencing and Navigation</i> , especifica como os objetos de aprendizagem devem ser sequenciados e qual a forma de navegação deles
LOOCV	A validação cruzada, <i>Leave-One-Out Cross Validation</i> , é um dos métodos de divisão do conjunto de dados
CDAE	O <i>Collaborative Denoising Auto-Encoder</i> é uma variação da arquitetura de redes neurais conhecida como <i>Auto Encoder</i>

Lista de símbolos

η	Taxa de aprendizagem da rede neural artificial
γ	<i>Momentum</i>
α	Camadas de uma arquitetura <i>Deep Auto Encoder</i>
k	k -ésima posição de um elemento
\approx	Aproximadamente
∇	Gradiente

Sumário

1	INTRODUÇÃO	25
1.1	Contexto	25
1.2	Aprendizado Adaptativo	26
1.3	E-learning	31
1.4	Problema	35
1.5	Motivação	35
1.6	Objetivo	36
1.7	Estrutura	37
I	FUNDAMENTOS	38
2	TRABALHOS RELACIONADOS	39
2.1	Visão Geral	39
2.2	Comparativo	44
2.2.1	Classificação	44
2.2.1.1	Seleção do Modelo de Estilo de Aprendizagem	45
2.2.1.2	Seleção de Descritores e Variáveis Alvo	47
2.2.1.3	Desenvolvimento do Modelo de Classificação e Avaliação	48
2.2.1.4	Trabalhos Relacionados	48
2.2.2	Recomendação	51
2.3	Conclusão	54
3	CONCEITOS FUNDAMENTAIS	55
3.1	Redes Neurais Artificiais	55
3.2	Multi Layer Perceptron	56
3.3	Deep Auto Encoder	58
3.4	Backpropagation	59
3.5	Descida do Gradiente Estocástico	61
3.6	Matriz Esparsa	62
3.7	Tipos de Aprendizagem de Máquina	63
3.8	Divisão do Conjunto de Dados	65
3.8.1	Estratégias para Divisão	65
3.8.2	Algoritmos para Divisão	66
3.9	Objetos de Aprendizagem	71
3.10	Modelo de Referência de Objeto de Conteúdo Compartilhável	72

3.11	Repositórios de Objetos de Aprendizagem	73
3.12	Estilos de Aprendizagem	74
II	CONTRIBUIÇÕES	76
4	MATERIAIS E MÉTODOS	77
4.1	Classificação	77
4.1.1	Conjunto de Dados	77
4.1.1.1	Variáveis Descritivas	78
4.1.1.2	Variáveis Alvo	79
4.1.2	Multi Layer Perceptron	82
4.1.3	Hiperparâmetros	83
4.1.3.1	Taxa de Aprendizado	83
4.1.3.2	Momentum	83
4.1.3.3	Quantidade de Épocas	84
4.1.3.4	Quantidade de Neurônios Ocultos	84
4.1.4	Divisão do Conjunto de Dados	84
4.1.5	Métricas de Avaliação	85
4.1.6	Treinamento da Rede	87
4.2	Recomendação	88
4.2.1	Conjunto de Dados	88
4.2.1.1	Objetos de Aprendizagem	91
4.2.1.2	Feedback	91
4.2.1.3	Análise dos Dados	92
4.2.1.4	Pré-processamento	93
4.2.2	Matriz Esparsa	93
4.2.3	Deep Auto Encoder	97
4.2.4	Hiperparâmetros	99
4.2.5	Divisão do Conjunto de Dados	100
4.2.6	Métricas de Avaliação	100
4.2.6.1	Online Evaluation	101
4.2.6.2	Accuracy	101
4.2.6.3	Recall	101
4.2.6.4	Normalized Discounted Cumulative Gain	102
4.2.6.5	Coverage	102
4.2.6.6	Mean Average Precision	103
4.2.7	Treinamento da Rede	103
5	RESULTADOS E ANÁLISES	105

5.1	Classificação	105
5.1.1	Conjunto de Dados de Treinamento	105
5.1.2	Agrupamentos	107
5.1.3	Modelo	109
5.1.4	Métricas	110
5.2	Recomendação	112
5.2.1	Divisão do Conjunto de Dados	112
5.2.2	Popularity Model	112
5.2.3	Deep Auto Encoder	113
5.2.3.1	Collaborative Denoising Auto Encoders	113
5.2.3.2	Deep Auto Encoder for Collaborative Filtering	116
5.2.3.3	Deep Auto Encoder for Collaborative Filtering With Content Information	118
5.2.4	Discussão	120
5.3	Discussão	122
6	CONCLUSÃO	124
6.1	Conclusões Gerais	124
6.2	Implicações Práticas	126
6.3	Trabalhos Futuros	127
	REFERÊNCIAS	128
	APÊNDICES	138
	APÊNDICE A – TRABALHOS PUBLICADOS PELO AUTOR	139
	APÊNDICE B – TRABALHOS AGUARDANDO A PUBLICAÇÃO	141
	APÊNDICE C – TAXONOMIA DE SISTEMAS DE RECOMEN- DAÇÃO	142
	APÊNDICE D – VISÃO GERAL DO EXPERIMENTO DE CLAS- SIFICAÇÃO	143
	APÊNDICE E – VISÃO GERAL DO EXPERIMENTO DE RECO- MENDAÇÃO	144

ANEXOS	145
ANEXO A – QUESTIONÁRIO PARA IDENTIFICAÇÃO DE ESTI- LOS DE APRENDIZAGEM	146

1 Introdução

Neste capítulo introdutório esta pesquisa é contextualizada, apresentando também os fundamentos sobre Aprendizado Adaptativo em sistemas de *e-learning*. Algumas formas de Aprendizado Adaptativo são descritas e discutidas. Os sistemas subjacentes, que suportam as estratégias de Aprendizado Adaptativo, são discutidos e definidos. O problema de Arquiteturas de Redes Neurais Artificiais aplicadas ao Aprendizado Adaptativo é formalizado e analisado dentro de diversos contextos. Os objetivos são apresentados e a estrutura do trabalho resumida.

1.1 Contexto

Aprendizado inteligente ou educação inteligente (*Smart Education*) inclui novos contextos educacionais nos quais a importância está focada no uso da tecnologia pelos alunos. Não depende apenas do *software* e *hardware* disponíveis, mas de como eles são articulados nas aulas ou no treinamento *on-line*; sendo um conceito subjacente ao conceito de Cidades Inteligentes (*Smart Cities*) (GOMEDE et al., 2018). A aprendizagem pode ser considerada uma atividade cerebral que envolve a aquisição, construção, organização, codificação, verificação, armazenamento e recuperação de informação (SCHUNK; MEECE; PINTRICH, 2012)¹. Desta forma, a habilidade de aprender pode ser interpretada como o gerenciamento de tal atividade. De acordo com Willingham (WILLINGHAM, 2019), seres humanos são naturalmente curiosos, mas não são naturalmente bons pensadores, em termos de raciocínio; a menos que as condições cognitivas estejam adequadas, eles evitarão raciocinar. Isto ocorre devido a três propriedades do pensamento. Primeiro, o raciocínio é lento. O sistema visual humano instantaneamente reconhece uma cena complexa, mas não é capaz de calcular da mesma forma um novo problema. Em segundo lugar, pensar é uma atividade que requer concentração para combinar novas informações com as existentes na memória de longo prazo. E, finalmente, o raciocínio é incerto porque erros são cometidos durante o processo e etapas precisam ser refeitas. Apesar destes aspectos, humanos gostam de pensar. Resolver problemas traz prazer, pois há uma sobreposição entre as áreas que contêm as substâncias químicas importantes na aprendizagem e aquelas importantes para o sistema de recompensa natural do cérebro (COLINO et al., 2020). O construtivismo afirma que a aquisição de conhecimento é o processo de construção baseado na experiência individual (SCHUNK; MEECE; PINTRICH, 2012). Além disso, tem sido reconhecido, em teorias psicológicas, que as experiências de aprendizagem e o conhecimento adquirido

¹ O aprendizado pode ser caracterizado como o procedimento pelo qual as informações são codificadas e armazenadas na memória de longo prazo (WILLINGHAM; HUGHES; DOBOLYI, 2015; WILLINGHAM, 2019).

são exclusivamente individuais. De acordo com a teoria da Gestalt ², os seres humanos podem gerar suas próprias experiências de aprendizado e interpretar as informações de diversas maneiras, pois cada pessoa tem uma perspectiva única do mundo (SCHUNK; MEECE; PINTRICH, 2012). Portanto, adaptar o conteúdo instrucional à preferência cognitiva individual pode ser uma das maneiras de melhorar a habilidade de raciocinar e, conseqüentemente, aprender, uma vez que raciocinar sobre algo é uma forma de aprendizado (WILLINGHAM; HUGHES; DOBOLYI, 2015).

Quando trata-se de aprendizagem, os termos personalização e adaptação, embora diferentes, são frequentemente utilizados de maneira intercambiável em diversos estudos (AROYO et al., 2006; GÖBEL; MEHM, 2013; GÓMEZ et al., 2014; LIN et al., 2013). Conceitualmente, a diferença entre ambos reside no fato de serem empregados para designar as diversas necessidades de aprendizado por meio de diferentes abordagens. Sem adaptar-se ao progresso contínuo da capacidade do aluno, o aprendizado personalizado pode ser alcançado por meio da identificação das características individuais. Além disso, o aprendizado adaptável pode ser implementado de acordo com o desempenho do aluno, sem identificar informações personalizadas relevantes, incluindo características e preferências individuais que podem afetar ainda mais seu progresso ou seu desempenho. No entanto, na prática, a semelhança entre aprendizado personalizado e aprendizado adaptativo está no fato da tecnologia adjacente suportar às diversas necessidades de aprendizado do aluno. Em particular, a fronteira entre aprendizado personalizado e adaptável se torna vaga ao limitar o escopo do aprendizado, em sistemas de *e-learning*, aprimorado pela tecnologia ³, pois foram usados como sinônimos em muitos estudos existentes (AROYO et al., 2006; GÖBEL; MEHM, 2013; GÓMEZ et al., 2014; LIN et al., 2013). Neste trabalho, ambos os termos foram simplificados para Aprendizado Adaptativo, pois existem mais similaridades entre eles do que disparidades; além disso, seus resultados práticos consistem em melhorar o desempenho dos alunos.

1.2 Aprendizado Adaptativo

Existem muitas formas de Aprendizado Adaptativo. Segundo Dockterman (DOCKTERMAN, 2018), cinco estratégias podem ser assumidas, da mais simples à mais complexa: (i) baseada no nome; (ii) auto descritiva; (iii) segmentada; (iv) baseada na cognição e (v) centrada no aluno. A primeira trata, simplesmente, de identificar as características descritivas do aluno, tais como o nome, por exemplo, e apresentar isto junto

² A psicologia da Gestalt é uma escola que surgiu na Áustria e na Alemanha no início do século XX, baseada no trabalho de Max Wertheimer, Wolfgang Köhler e Kurt Koffka. Os psicólogos enfatizaram que os organismos percebem padrões integralmente, não apenas componentes individuais (COUDRAY, 2020).

³ *Technology Enhanced Learning* (TEL) é uma área da educação que se ocupa do estudo das tecnologias sobre o aprendizado.

ao conteúdo. A segunda, coleta mais informações, fornecidas pelo próprio aluno, como preferências de conteúdos, formatos, entre outros. Esta estratégia adapta, em certo nível, o conteúdo instrucional de acordo com as preferências informadas. A terceira procura agrupar os alunos por similaridade fornecendo a instrução adaptada ao grupo. Esta estratégia pode auxiliar em trabalhos realizados em grupos pela combinação de diferentes tipos de preferências dos alunos. A quarta analisa as preferências relacionadas à cognição, como por exemplo, a escolha de um áudio ao invés de um texto. Por fim, a opção centrada no aluno armazena dados sobre o desempenho em avaliações para melhorar seus resultados. Além disso, esta estratégia pressupõe a adaptação em tempo real ajustando-se ao nível de aprendizagem do aluno. Independente da estratégia, ou combinações de estratégias, existe a necessidade de um sistema que as suportem (BERNARD et al., 2017). Geralmente, estes sistemas adjacentes são computacionais e baseados em Tecnologia da Informação & Comunicação (TIC) (ROMERO; VENTURA, 2010; NORMADHI et al., 2019).

Uma das formas de Aprendizado Adaptativo, de acordo com Felder e Silverman (FELDER; SILVERMAN et al., 1988), está relacionada com a identificação dos estilos de aprendizagem, que são parte do estilo cognitivo ⁴ (GOMEDE; BARROS; MENDES, 2020) sendo, portanto, associada à estratégia (iv) baseada na cognição. Estes estilos descrevem as preferências dos alunos sobre como o conteúdo é apresentado, trabalhado e internalizado (NAFEA; SIEWE; HE, 2019). De acordo com Willingham (WILLINGHAM; HUGHES; DOBOLYI, 2015; WILLINGHAM, 2019), os alunos podem apresentar diversas maneiras preferidas de aprender. Portanto, conhecer o estilo de aprendizagem do aluno pode ser fundamental para encontrar a maneira adequada de melhorar o processo de aprendizagem. Estudos mostram que os estilos de aprendizagem permitem que os sistemas de *e-learning* adaptáveis melhorem o processo de aprendizagem dos alunos (KOLB; KOLB, 2005; BERNARD et al., 2017; NORMADHI et al., 2019; NAFEA; SIEWE; HE, 2019; COLINO et al., 2020). Isto se deve ao fato de que as restrições e preferências de cada aluno são respeitadas e o progresso pode ser monitorado e assistido de forma individualizada. Além disso, permite com que atuações sejam feitas sempre que algum desvio ocorra, permitindo que professores, tutores, administradores e/ou pais possam ter uma visão melhor dos resultados obtidos (GOMEDE et al., 2018). Por exemplo, adaptar o conteúdo de acordo com os estilos de aprendizagem dos alunos melhorou os resultados do aprendizado e diminuiu o tempo de aprendizado (NORMADHI et al., 2019). Pesquisas também apontam que a aprendizagem dos alunos melhoram quando o estilo de ensino dos instrutores correspondem aos estilos de aprendizagem dos alunos (BERNARD et al., 2017).

Segundo Willingham (WILLINGHAM, 2019), a previsão de qualquer estilo de

⁴ Os estilos cognitivos foram definidos como diferentes preferências individuais de organização e processamento de informações e experiências, enquanto os estilos de aprendizagem são habilidades e preferências individuais que afetam a maneira como os alunos percebem, reúnem e processam materiais de aprendizagem (SADLER-SMITH, 2001).

aprendizagem representa que um método de ensino é adequado para alguns alunos, mas não para outros. Portanto, é possível se beneficiar de diferentes tipos de aprendizado. É importante entender a diferença entre capacidade de aprendizagem e estilo de aprendizagem. Capacidade de aprendizagem é a habilidade ou o sucesso de certos tipos de pensamento (matemática, por exemplo). Em contraste com as habilidades, o estilo de aprendizagem é a tendência de se pensar de uma maneira particular, por exemplo, pensar sequencial ou holisticamente, o que é independente do contexto (WILLINGHAM, 2019). Há uma contraposição entre a popularidade dos estilos de aprendizagem na educação e as evidências de sua validade. De acordo com Pashler (PASHLER et al., 2008), a validade razoável do uso de estilos de aprendizagem para melhorar o aprendizado dos alunos ainda deve ser demonstrada. No entanto, em seu estudo, Chang (CHANG; HUNG; LIN, 2015) examinam os recentes avanços da pesquisa sobre aprendizado experimental no ensino superior e analisam como ele pode melhorar o aprendizado dos alunos. Em seus estudos, eles concluem que os estilos de aprendizagem podem se basear na pesquisa e na observação clínica dos padrões dos estilos de aprendizagem e podem ser aplicados em todo o ambiente educacional, por meio de um programa de desenvolvimento institucional.

Além disso, como indicado por Willingham (WILLINGHAM; HUGHES; DOBOLYI, 2015), psicólogos dispõem de poucas abordagens para testar a preferência de aprendizado, levando, portanto, a algumas hipóteses. Primeiro, considera-se que o estilo de aprendizagem é estável para um indivíduo. Em outras palavras, se um aluno tem um estilo de aprendizado específico, este estilo deve ser parte estável da composição cognitiva do aluno. Segundo, o estilo de aprendizado deve ser consequente; desta forma, o uso de um estilo de aprendizado específico deve ter implicações nos resultados do aprendizado dos alunos. Assim, uma teoria de estilos de aprendizagem deve ter três características; (i) um estilo de aprendizagem específico deve ser uma característica estável de uma pessoa; (ii) indivíduos com estilos diferentes devem pensar e aprender de maneiras diferentes; e (iii) indivíduos com estilos diferentes não diferem, em média, em sua capacidade (WILLINGHAM, 2019). Desta forma, ao utilizar algum modelo de estilo de aprendizagem, este modelo deve ser eficiente para suportar tais hipóteses e, assim, ter uma melhor maneira de prever as preferências do aluno.

De acordo com Bernard (BERNARD et al., 2017), existem vários modelos para identificar os estilos de aprendizagem. Um dos mais conhecidos é o Modelo de Estilos de Aprendizagem Felder-Silverman (FSLSM). Este modelo propõe quatro dimensões para classificar os estilos de aprendizagem, com cada dimensão sendo representada por um intervalo discreto e fechado de 0 a 11 ($[0 : 11]$), em ambas direções (FELDER; SILVERMAN et al., 1988). A primeira dimensão deste modelo, chamada de **Processamento** e composta das preferências **Ativo/Reflexivo**, determina se o aluno prefere primeiro experimentar (Ativo) sobre um assunto para então, raciocinar sobre ele (Reflexivo). A segunda dimensão, chamada de **Percepção** é composta das preferências **Sensitivo/Intuitivo**, determina se

o aluno prefere tocar em coisas para aprender (Sensitivo) ou observar fenômenos para induzir informações (Intuitivo). A terceira dimensão, **Input**, contendo **Visual/Verbal**, determina se o aluno prefere ver gráficos, tabelas, figuras, *outlines*, simulações (Visual) em vez de ler ou ouvir textos (Verbal)⁵. Finalmente, a dimensão chamada de **Entendimento** e composta das preferências **Sequencial/Global** determina se o aluno prefere obter as informações de maneira sucessiva, aprendendo passo a passo (Sequencial) ou obter uma visão geral das informações primeiro e depois ir aos detalhes, sem uma sequência predefinida (Global). Estas quatro dimensões não são deterministas, mas sim, indicam a tendência de preferência do aluno para o recebimento, processamento e armazenamento de informações (FELDER; SILVERMAN et al., 1988; BERNARD et al., 2017; NORMADHI et al., 2019). Maiores detalhes serão tratados na Seção 3.12.

Tradicionalmente, os estilos de aprendizagem são medidos, principalmente, por meio de pesquisas e questionários, nos quais os alunos são solicitados a auto-avaliar seus próprios comportamentos (BERNARD et al., 2017). No entanto, esta abordagem apresenta algumas limitações. Primeiro, a interferência externa pode atrapalhar os resultados durante sua aplicação. Segundo, os resultados são influenciados pela qualidade da pesquisa ou questionário. E, finalmente, diferentes alunos podem interpretar as perguntas de uma maneira diferente (WILLINGHAM, 2019; NORMADHI et al., 2019). De acordo com Bernard (BERNARD et al., 2017), a caracterização dos estilos de aprendizagem é um problema que lida com múltiplos descritores e determinadores⁶. Os descritores podem surgir de várias fontes, como *logs* de sistemas de *e-learning*, questionários e/ou bancos de dados. Além disso, os descritores, geralmente, estão associados a objetos de aprendizagem como fóruns, conteúdos, *outlines*, questionários, auto-avaliações, exemplos e outros tipos de recursos. Os determinadores são utilizados para permitir a classificação do estilo de aprendizagem como resultado de uma combinação de descritores, que podem indicar se um aluno deve ser classificado como Ativo/Reflexivo, Sensitivo/Intuitivo, Visual/Verbal e Sequencial/Global, com base em sua preferência para reconhecer, processar e armazenar informações (NORMADHI et al., 2019; WILLINGHAM, 2019).

Outra forma de Aprendizado Adaptativo está relacionada com a recomendação de objetos de aprendizagem adequados ao aluno e tem sido proposta por diversos pesquisadores com diferentes abordagens (TANG; MCCALLA, 2005; BOBADILLA; SERRADILLA; HERNANDO, 2009; DRACHSLER et al., 2009; KLAŠNJA-MILIĆEVIĆ et al., 2011; MANOUSELIS et al., 2011; VESIN et al., 2013; BUDER; SCHWIND, 2012; DRACHSLER et al., 2015; KRAUSS, 2016; HARUNA et al., 2017; BARBIERI et al., 2017; NAFEA; SIEWE; HE, 2019). Este tipo de Aprendizado Adaptativo identifica o comportamento do

⁵ Até o presente momento, não existem evidências significativas de que uma determinada dimensão não possa ser aplicada a determinadas habilidades. Por exemplo, **visual** para habilidades em áreas exatas e **verbal** para áreas de letras.

⁶ Descritores e determinadores, em termos computacionais, são considerados como variáveis independentes e dependentes para a construção de um modelo.

aluno, geralmente utilizado métodos como filtragem colaborativa, para recomendar um conjunto de objetos de aprendizagem que possam ser úteis e aumentem o desempenho do aluno. Estes objetos são compostos de parte do conteúdo instrucional que podem ser adaptados, em termos de **formato** e **conteúdo**, para fornecer uma melhor experiência de aprendizagem. Além disso, estes objetos permitem o reuso, compartilhamento, gerenciamento e a padronização ao adotar modelos de construção. A recomendação de objetos de aprendizagem é uma estratégia mais granular de adaptação se comparada com a de estilos de aprendizagem. Enquanto a estratégia de estilo de aprendizagem tem como objetivo classificar o aluno de acordo com preferências pré-definidas, um sistema de recomendação identifica a preferência exclusiva de um determinado aluno e faz as recomendações para esta preferência (NAFEA; SIEWE; HE, 2019). Além disso, este tipo de estratégia permite explicar o comportamento do aluno em termos de preferência de objetos de aprendizagem e prever comportamentos futuros do mesmo (BARBIERI et al., 2017). O Aprendizado Adaptativo, por meio de sistemas de recomendação no contexto educacional, exige certos requisitos para ser efetivo. Dentre estes requisitos estão a interação responsável entre **produtor/consumidor** e o conjunto de objetos que podem ser recomendados. Maiores detalhes serão tratados na Seção 2.1.

Desta forma, um ambiente de aprendizagem pode ser considerado adaptativo caso seja capaz de monitorar as atividades do aluno interpretando-as com base em dados; inferir requisitos e preferências das atividades representando-as adequadamente em modelos; e, finalmente, atuar sobre o conhecimento disponível do aluno e do assunto em questão, para facilitar, dinamicamente, o processo de aprendizagem (PARAMYTHIS; LOIDL-REISINGER, 2003; NAFEA; SIEWE; HE, 2019). Portanto, aprendizado adaptativo desempenha um importante papel na melhoria dos resultados obtidos pelos alunos. Este tipo de abordagem permite que elementos pedagógicos como controle de fluxo, sequenciamento de atividades, aprendizado adaptativo para corrigir determinados desvios, engajamento e atenção, além de recomendações adaptadas, possam ser considerados como fatores a contribuir com tais resultados (MOUBAYED et al., 2018). Desta maneira, a aprendizagem adaptativa, suportada por tecnologia (TIC), pode ser considerada uma área importante de pesquisa na área de tecnologias educacionais (NAFEA; SIEWE; HE, 2019). Adicionalmente, os sistemas de aprendizado adaptativo são construídos com base no desenvolvimento de mídias adaptativas e definiram o aprendizado como, geralmente, se referindo ao uso de sistemas e/ou ferramentas de computador para fornecer materiais e/ou atividades de aprendizado sob medida para atender as necessidades de aprendizado (BRUSILOVSKY; KARAGIANNIDIS; SAMPSON, 2004; HARUNA et al., 2017). Tais softwares são sistemas inteligentes de tutoria, que empregam algoritmos ou modelos computacionais para fornecer retroalimentação imediata e instruções de aprendizado para o aluno, sem a intervenção de professores (PSOTKA; MASSEY; MUTTER, 1988; NAFEA; SIEWE; HE, 2019). Estes sistemas pertencem a categoria chamada *e-learning systems* (NORMADHI et al., 2019).

1.3 E-learning

O emprego de TIC possibilita a construção de ambientes para Aprendizado Adaptativo onde o aluno pode ser identificado, o conteúdo pode ser especializado e o progresso monitorado, suportado e assistido (ROMERO; VENTURA, 2007; ROMERO; VENTURA, 2010; NORMADHI et al., 2019). Por exemplo, dois alunos podem receber a mesma instrução simultaneamente, sendo que cada um recebe um conjunto de Objetos de Aprendizagem completamente diferente. Um dos grandes benefícios de um sistema de Aprendizado Adaptativo é realizar instruções complexas de maneira adequada às preferências do aluno. Além disso, quando um conteúdo é entregue de forma correta, o sistema pode aprender mais sobre o comportamento e perfil do aluno. Nos últimos anos, o Aprendizado Adaptativo tem sido realizado por meio de métodos disponibilizados pelo desenvolvimento da TIC (DAWSON; HEATHCOTE; POOLE, 2010; MACFADYEN; DAWSON, 2012). A aprendizagem adaptável ⁷ tornou-se possível por meio de sistemas inteligentes, que integram as preferências do aluno aos dados de aprendizagem individualizada ⁸ sendo que os objetivos de aprendizado, as abordagens, o conteúdo instrucional e seu sequenciamento podem variar de acordo com as necessidades individuais (ROMERO; VENTURA, 2010; BODILY; VERBERT, 2017; MOUBAYED et al., 2018; SHEEBA; KRISHNAN, 2019; CHANG; LIN; CHEN, 2019; WILLINGHAM, 2019; COLINO et al., 2020). Portanto, o aprendizado adaptativo fornece a instrução de forma que o ritmo e a abordagem instrucional são otimizados para as necessidades de cada aluno e suportado por sistemas de *e-learning*.

Dentre os exemplos da utilização de TIC em sistemas de *e-learning*, pode-se citar o modelo de aprendizado conhecido como Cursos Online Abertos e Massivos (*Massive Open Online Course*, MOOC). Este modelo é adotado por universidades e aplicado para permitir o aprendizado bidirecional por meio de mídias sociais (BARAK; WATTED; HAICK, 2016; EL-HMOUDOVA; LOUDOVA, 2018; YU et al., 2018). Os MOOC vêm desempenhando um importante papel para a aprendizagem (ROMERO; VENTURA, 2007; ROMERO; VENTURA, 2010; PEÑA-AYALA, 2014; LU et al., 2015; YU et al., 2017; RODRIGUES; ISOTANI; ZÁRATE, 2018; LAI; BOWER, 2019). Um dos fatores para isso está relacionado à possibilidade dos alunos acessarem os conteúdos de maneira adaptada e estudar de acordo com suas preferências. Portanto, o Aprendizado Adaptativo é um ponto fundamental para que um MOOC possa melhorar o aprendizado e permitir que demais agentes como professores, tutores, pais e administradores possam ter uma visão de todo processo de aprendizagem (GOMEDE et al., 2018). Entretanto, pesquisadores apresentaram

⁷ Os termos adaptação e personalização serão utilizados como sinônimos neste trabalho. Apesar de serem conceitualmente diferentes, o resultado prático é o mesmo: adaptar-se às preferências e comportamentos do aluno para melhorar seus resultados.

⁸ Aprendizado Adaptativo utilizando computação, segundo Willingham, é uma solução nova para um problema antigo: mais alunos do que professores (WILLINGHAM, 2019).

estudos sobre como o formato MOOC afetou o ensino superior (ALRAIMI, 2015; TOVENLINDSEY; RHOADS; LOZANO, 2015). Friedman (FRIEDMAN; FRIEDMAN, 2013) argumentou que os MOOC representam uma evolução na aprendizagem universitária, mas ainda não está claro se o ensino baseado em tecnologia web pode manter os níveis de atenção dos alunos e melhorar a eficácia da aprendizagem (MARGARYAN; BIANCO; LITTLEJOHN, 2015). Adicionalmente, como cita Chang (CHANG; LIN; CHEN, 2019), a virtualização em sala de aula também gerou problemas, como a distração promovida por ferramentas contidas nos sistemas de *e-learning*.

Outros exemplos do emprego de TIC, em sistemas de *e-learning*, podem ser vistos nos trabalhos de Wu (WU et al., 2012), que propõem um sistema adaptativo de aprendizagem de vocabulário em inglês baseado na teoria de resposta a itens e no ciclo de memória de aprendizagem que recomendava a aquisição adequada de vocabulário. Este estudo fornece uma estrutura de aprendizado adaptativo, considerando a teoria dos ciclos de memória de Hermann Ebbinghaus e a capacidade de vocabulário do aluno. Shih (SHIH; CHUANG; HWANG, 2010) propõem um algoritmo heurístico para determinar caminhos de aprendizagem adaptada, suportado por um sistema de *e-learning* sensível ao contexto, considerando vários fatores, incluindo a relevância do objeto de aprendizagem, a significância dos caminhos e o número de visitantes simultâneos. Outras abordagens foram propostas como a de Chen (CHEN; LEE; TSAI, 2011), onde um sistema adaptado fornece caminhos de aprendizagem que podem ser adaptados a vários níveis de dificuldade dos materiais e habilidades do aluno. Adicionalmente, Klavnsnja (KLAŠNJA-MILIĆEVIĆ et al., 2011) afirmam que o aprendizado adaptativo ocorre quando os sistemas de *e-learning* fazem esforços deliberados para projetar experiências educacionais que atendem às necessidades, objetivos, talentos e interesses do aluno. Também, pode-se citar, a utilização de mineração de texto para identificação de termos mais utilizados e a análise de sentimentos (MOUBAYED et al., 2018; ROMERO; VENTURA, 2010; DUTT; ISMAIL; HERAWAN, 2017).

De forma adicional, Hwang e Chang (HWANG; CHANG, 2011) utilizaram vários recursos adaptativos, incluindo as quatro dimensões do estilo de aprendizagem de Felder-Silverman e o modelo de estilo cognitivo dependente/independente do campo, para desenvolver um sistema de aprendizagem adaptável. Posteriormente, por meio dos experimentos em duas turmas de estudantes de graduação, os resultados de seus estudos apontaram a eficácia da estrutura de aprendizagem adaptativa proposta (HWANG; YANG; WANG, 2013). Mais recentemente, Hwang (HUANG et al., 2016) desenvolveu um sistema móvel e ubíquo de aprendizado de vocabulário adaptado, empregando a semelhança semântica entre o vocabulário alvo e os contextos do local onde o mesmo seria utilizado. Os experimentos deste estudo (HUANG et al., 2016) revelaram que os resultados, tanto do desempenho quanto da motivação, melhoraram no grupo experimental de estudantes que usaram o sistema comparado aos materiais de aprendizagem convencionais. Adicionalmente,

Chang (CHANG; HUNG; LIN, 2015) expandiu o modelo de processamento de informações proposto por Atkinson e Shiffrin (MILLER et al., 1987) e definiu o aprendizado como uma atividade psicológica interna que ocorre via consciência sensorial, atenção, reconhecimento, transformação, memória e outras ações, bem como os processos de absorção e aplicação de conhecimento. De acordo com este modelo, processos cognitivos individuais se referem à percepção, memória e pensamento, enquanto o processo secundário de percepção se refere ao reconhecimento e atenção de padrões. Portanto, atenção e cognição estão relacionadas e ligadas (CHANG; LIN; CHEN, 2019). Este trabalho utiliza tais conceitos sobre um MOOC. Todos os trabalhos citados anteriormente se baseiam no uso de TIC para a construção de sistemas adaptativos de *e-learning*.

Portanto, o avanço da TIC permitiu a utilização destes recursos em larga escala por meio dos sistemas de *e-learning*. Tais sistemas têm como finalidade suportar diferentes tipos de usuário e não somente os alunos. Neste caso, alguns deles são (i) alunos, (ii) professores, (iii) pesquisadores, (iv) administradores, (v) governo e (vi) pais ou responsáveis (GOMEDE et al., 2018). Algumas maneiras para obter personalização e adaptação nesses sistemas são via utilização de técnicas de inteligência computacional aliadas a aquisição de dados que representam o comportamento dos alunos durante as atividades de aprendizagem. Esta combinação de TIC e inteligência computacional para Aprendizado Adaptativo em sistemas de *e-learning* permite o *feedback*⁹ para seus usuários. Isto é importante para os seres humanos, funcionando como um equipamento que fornece as coordenadas de direção correta (NORMADHI et al., 2019).

Pode-se considerar um sistema adaptativo de *e-learning* sendo composto de duas estruturas fundamentais que permitem o Aprendizado Adaptativo (MOUBAYED et al., 2018): (i) aquisição de dados e (ii) algoritmos de inteligência computacional. Para a aquisição de dados, existem duas categorias, distintas e complementares, que buscam representar o comportamento dos alunos ao trabalharem com os objetos oferecidos pelo sistema; (i) *feedback implícitos* e (ii) *feedback explícitos* (BODILY; VERBERT, 2017). O primeiro grupo captura dados sem o conhecimento do aluno e/ou a necessidade deste, eventualmente, realizar uma ação para isso. Alguns destes tipos de *feedback* são representados pela quantidade de visualização de certos objetos, tempo gasto nestes objetos, entre outros. O segundo grupo coleta os dados utilizando ações do aluno, de maneira consciente. Exemplos desse grupo podem ser comentários feitos no objeto, se o objeto foi marcado para notificação futura e/ou até mesmo se o objeto foi adicionado como favorito na lista do aluno. Estes objetos que são utilizados para a aprendizagem são conhecidos como Objetos de Aprendizagem e são empregados tanto em um sistema de *e-learning* quanto em ambientes tradicionais (BODILY; VERBERT, 2017). Exemplos destes objetos estão relacionados com vídeos, simulações, questionários, atividades auto avaliativas, fóruns de

⁹ O termo *feedback*, neste trabalho, está sendo considerado como uma resposta imediata a uma ação de um determinado aluno.

discussão, livros, artigos, entre outros. A forma como o aluno interage com esses objetos influencia nos dados adquiridos (DRACHSLER et al., 2015).

Neste sentido, os algoritmos de inteligência computacional permitem a utilização dos dados adquiridos para o processamento e, conseqüentemente, o Aprendizado Adaptativo (MOUBAYED et al., 2018). Existem diversas finalidades para estes algoritmos que vão desde a classificação de estilos de aprendizagem até a recomendação de OA. Dentre os tipos de algoritmos utilizados, pode-se destacar as redes neurais artificiais, nas mais variadas arquiteturas, para classificação e recomendação (WANG; WANG; YEUNG, 2015). Além disso, existem algoritmos baseados em árvores que possibilitam a construção de uma hierarquia de características para adicionar o aluno a uma determinada classe pré-estabelecida. Estes algoritmos estão relacionados com a identificação de estilos de aprendizagem (BERNARD et al., 2017). Da mesma forma, existem algoritmos que permitem o agrupamento de vários alunos, tentando identificar os que possuem as características similares para permitir um Aprendizado Adaptativo baseado em grupos (NORMADHI et al., 2019). Além disso, pode-se citar a aplicação de redes complexas para entender o comportamento em rede, sobre como certas informações serão propagadas pelo aluno dentro do sistema de *e-learning* (GAO; BARZEL; BARABÁSI, 2019). E ainda, pode-se citar a utilização destes algoritmos para a análise e visualização de dados, fornecer *feedback* a professores e administradores, prever o desempenho, modelar as preferências do aluno, detectar comportamentos indesejáveis, desenvolver mapas conceituais, construir cursos e realizar o planejamento e agendamento (ROMERO; VENTURA, 2007; ROMERO; VENTURA, 2010).

Dentre as redes neurais artificiais, existem dois tipos de arquitetura que vem ganhando destaque para sistemas de Aprendizado Adaptativo em *e-learning*, a **Multi Layer Perceptron** (MLP) e a **Deep Auto Encoder** (DAE) (ZUO et al., 2016; WANG; WANG; YEUNG, 2015; BERNARD et al., 2017; MONGIA et al., 2020; KIRAN; KUMAR; BHASKER, 2020; NASSAR; JAFAR; RAHHAL, 2020; CHEN et al., 2020). A primeira suporta diversas camadas intermediárias, múltiplas saídas e utiliza um aprendizado supervisionado baseado no algoritmo *backpropagation* (Seção 3.4). A segunda possui um aprendizado semi supervisionado, onde o conjunto de saídas da rede $X' = \{1, 2, \dots, n\}$ é comparado ao conjunto de entradas $X = \{1, 2, \dots, n\}$; ou seja, o número de elementos no conjunto de entrada é o mesmo do conjunto de saída. Desta forma, não existe a necessidade de um conjunto de dados rotulados para que a aprendizagem da rede seja feita (WANG; WANG; YEUNG, 2015). Este tipo de arquitetura permite a fatoração de matrizes, criando um espaço latente menor do que a entrada (na etapa de codificação) e restaurando este espaço na saída da rede (na etapa de decodificação). Isto permite o processamento de matrizes esparsas que são comuns em sistemas de recomendação baseados em filtragem colaborativa (WANG; WU, 2011; WANG; WANG; YEUNG, 2015). Além disso, esta arquitetura de rede utiliza os mesmos conceitos de treinamento, como o

algoritmo de *backpropagation* e a descida do gradiente estocástico (*Stochastic Gradient Descent*) (BOTTOU, 2010). Possui também a possibilidade de correção de ruídos, o que é muito relevante em sistemas de Aprendizado Adaptativo que possam ter *feedback* feitos por descuido (KRAUSS, 2016). Independente da estratégia utilizada, um dos pontos centrais está relacionado à definição do problema.

1.4 Problema

Neste contexto, para alcançar o Aprendizado Adaptativo, pode-se classificar o aluno em um estilo de aprendizagem e, de forma mais **granular**, ajustar a sua preferência individual para receber objetos de aprendizagem. O primeiro problema exige um conjunto de descritores \mathbf{D} e um modelo de estilo de aprendizagem M . Além disso, exige-se um algoritmo capaz de utilizar os descritores para **classificar** o aluno. O problema de **recomendar** objetos de aprendizagem para um aluno, em um sistema de *e-learning*, pode ser descrito como: dada uma matriz esparsa ($A \times O$), onde o \mathbf{A} representa o vetor de alunos que interagiram com os objetos de aprendizagem, representados pelo vetor \mathbf{O} , por meio de *feedback* implícitos e explícitos, retorne uma lista de objetos, contendo um subconjunto de \mathbf{O} , \mathbf{O}' com a maior probabilidade de um novo *feedback*. Este novo *feedback* tem por finalidade melhorar a recomendação de próximos objetos adaptando as preferências do aluno. Sendo, portanto, o resultado da recomendação de uma lista ordenada dos objetos de aprendizagem que possam ser utilizados pelo aluno em um determinado contexto para melhorar sua aprendizagem. De forma geral, o **problema de Aprendizado Adaptativo** pode ser definido como:

Definição 1 *Adapte o conjunto de preferências \mathbf{P} de um aluno A baseado em sua interação com o conjunto de objetos de aprendizagem \mathbf{O} .*

onde \mathbf{P} representa o estilo de aprendizagem ou a lista de objetos de aprendizagem adequados ao aluno A , e \mathbf{O} representa o conjunto de interações entre aluno/objeto. Este conjunto de interações possui os descritores do problema.

1.5 Motivação

Além dos benefícios já citados pelos demais autores, existem algumas motivações para a resolução deste problema. Primeiro, do ponto de vista educacional, isto pode representar um avanço na forma como a educação é desempenhada pelos agentes (GOMEDE et al., 2018). Pode trazer, também, resultados importantes em médio e longo prazo para um país (GAFFO et al., 2017). Segundo, este problema permite explorar conceitos de neurociência aplicados a aprendizagem e suportados pela computação (GOMEDE;

BARROS; MENDES, 2020). Terceiro, este problema explora o estado da arte em termos de inteligência computacional, especificamente arquiteturas de redes neurais artificiais, para resolver o problema de múltiplos descritores e de classificação e recomendação de dados não lineares e esparsos. Portanto, além das motivações computacionais e educacionais, podem-se citar as implicações práticas para os sistemas de *e-learning*. Por fim, este problema contribui para duas grandes áreas de pesquisa atuais, neurociência e computação.

1.6 Objetivo

Esta pesquisa tem como objetivo investigar o uso de algoritmos de inteligência computacional (IC), conhecidos como redes neurais artificiais, para analisar e melhorar a precisão do uso de abordagens autônomas de Aprendizado Adaptativo em sistemas de *e-learning*. A hipótese de trabalho pode ser descrita como: **se o padrão de aprendizagem de um aluno puder ser reconhecido então pode-se descrever e prever seu comportamento em um sistema de *e-learning***. Portanto, parte do possível padrão de aprendizagem de um aluno está relacionado com a forma e o conteúdo que este tem maior predisposição à aprendizagem. Assim, realizou-se esta pesquisa para identificar os recursos que podem representar as preferências de um aluno com base em informações massivas (*Big Data*) coletadas em um MOOC, dentro de um sistema de *e-learning*, e usar estes recursos para Aprendizado Adaptativo. Além disso, investigou-se o uso das arquiteturas de rede neurais artificiais *Multi Layer Perceptron* e *Deep Auto Encoder* para classificar os estilos de aprendizagem e recomendar objetos de aprendizagem, respectivamente. Por fim, investigou-se técnicas de hiperparâmetrização e conceitos de neurociências associados à aprendizagem.

Além do objetivo principal, alguns objetivos secundários são investigados como forma de suportar o alcance do objetivo principal.

- Avaliar um conjunto de hiperparâmetros para arquiteturas de redes neurais artificiais
- Avaliar objetos de aprendizagem baseadas no modelo SCORM
- Avaliar o comportamento dos alunos baseado na interação com objetos de aprendizagem
- Avaliar um conjunto de *feedback* implícitos e explícitos para *e-learning*
- Avaliar o problema de cauda longa gerada por aluno/objeto com pouca interação
- Avaliar o problema de partida fria ocasionada por aluno/objeto sem interação
- Avaliar a cobertura para gerar recomendações do maior número de objetos

- Avaliar o *overfitting* ao recomendar objetos muito similares aos já recomendados
- Avaliar o problema de alunos com comportamento atípico
- Avaliar o problema com novo objeto adicionado sem nenhum *feedback*
- Avaliar os modelos de estilos de aprendizagem
- Avaliar os repositórios de objetos de aprendizagem
- Avaliar as estratégias para divisão do conjunto de dados

1.7 Estrutura

A estrutura deste trabalho é composta, além desta introdução, do Capítulo 2 contendo os trabalhos relacionados que abordaram o assunto, fornecendo uma comparação entre eles e este. O Capítulo 3 com os conceitos necessários para a fundamentação do trabalho. O Capítulo 4 com materiais e métodos, descrevendo, de forma detalhada, todos os componentes necessários para a criação de modelos, baseado nas redes neurais artificiais, para classificação e recomendação, visando o Aprendizado Adaptativo do sistema de *e-learning* às preferências do aluno. O Capítulo 5 com as análises e discussões dos resultados do experimentos realizados. E, por fim, um capítulo com as conclusões, consequências teóricas e práticas e recomendações para trabalhos futuros. Adicionalmente, o Anexo A fornece a lista de trabalhos publicados pelo autor, durante a pesquisa. O Anexo B fornece a lista de trabalhos a serem publicados pelo autor com resultados adicionais à pesquisa. O Anexo C apresenta uma taxonomia de tipos de sistemas de recomendação que podem ser utilizados para Aprendizado Adaptativo. O Anexo D apresenta uma visão geral do experimento de classificação. Já o Anexo E, apresenta uma visão geral do experimento de recomendação. Por fim, o Anexo A apresenta as perguntas do questionário (FSLSM) para identificação de estilos de aprendizagem e um experimento realizado.

Parte I
Fundamentos

2 Trabalhos Relacionados

Este capítulo apresenta os trabalhos relacionados com a utilização de arquiteturas de redes neurais artificiais para Aprendizado Adaptativo em sistemas de *e-learning*. Os trabalhos foram organizados em dois grupos: (i) trabalhos relacionados com questões computacionais que apresentem os fundamentos de arquiteturas de redes neurais, e que foram utilizados para a construção dos experimentos; e (ii) os trabalhos que utilizam arquiteturas de redes neurais para Aprendizado Adaptativo em sistemas de *e-learning* e apresentem conceitos relacionados com educação e aprendizagem. Os tipos mais frequentes de Aprendizado Adaptativo, como classificação e recomendação, foram utilizados. Além disso, este capítulo serve como base para comparação dos resultados os demais autores com os desta pesquisa no Capítulo 5.

2.1 Visão Geral

O Aprendizado Adaptativo em sistemas de *e-learning* é, geralmente, aplicado a um grande número de tarefas (ROMERO; VENTURA, 2010; NORMADHI et al., 2019). Modelagem de preferências, analisar e visualizar dados, fornecer *feedback* para instrutores, prever o desempenho dos alunos, detectar comportamentos não desejados, agrupar alunos, análise de redes sociais, desenvolver mapas conceituais, construção de programas, planejamento, agendamento, classificação de alunos e recomendação de objetos de aprendizagem são alguns exemplos. No geral, tarefas de **classificação** são bem definidas e aplicam-se a identificar as preferências de um aluno (BERNARD et al., 2017). Entretanto, sistemas de **recomendação** são, geralmente, aplicados a um contexto de negócio com o objetivo de recomendar produtos que sejam da preferência dos usuários e com a finalidade de aumentar as vendas. Desta forma, este tipo de recomendação é baseado em preferências e gostos. Por outro lado, um sistema de recomendação, no contexto educacional, possui alguns requisitos que devem ser considerados para serem aplicados com efetividade (BUDER; SCHWIND, 2012). Vários sistemas de recomendação em *e-learning* foram projetados para recomendar um grande número de elementos como lições de idiomas, artigos, livros, vídeos, objetos de aprendizagem, notas de leitura ou até mesmos cursos completos. Portanto, a finalidade principal de um sistema de recomendação, no contexto educacional, é permitir o Aprendizado Adaptativo útil para alcançar as necessidades de aprendizagem do aluno (BARBIERI et al., 2017; KRAUSS, 2016).

Projetar e construir um sistema de recomendação educacional pode apresentar vários desafios. Além das questões computacionais envolvidas, em que precisa-se armazenar as possíveis interações da combinação aluno/objeto, gerando um substancial volume de

dados, uma quantidade considerável de interações aluno/objeto é necessária para que o resultado seja satisfatório. Se a quantidade de alunos que geram tais dados for pequena, a recomendação torna-se menos precisa. Além disso, quando o aluno não possui nenhum dado armazenado no sistema, o recomendador irá oferecer uma lista de objetos mais populares; isto é conhecido como o problema de partida fria (*cold-start problem*). Com isto, o aluno seleciona alguns dos objetos como interessante e outros como não. Desta forma, este objeto fica disponível para que outros alunos, que sejam similares, possam se beneficiar da recomendação sem a necessidade de conhecer os demais. Portanto, este ciclo entre atividades individuais (**selecionar e avaliar**) e de sistema (**agrupar e filtrar**) fornece os cinco pilares aos quais os sistemas de recomendação estão apoiados (VESIN et al., 2013).

O primeiro pilar está relacionado com a filtragem colaborativa, ou seja, depende da **interação responsável** de alunos com objetos de aprendizagem para que o sistema seja útil. Por exemplo, a recomendação de um livro pode ser feita de forma equivocada, por um conjunto de alunos, ao gerar um evento de avaliação de forma descuidada. Em contrapartida, a recomendação pode ser feita por meio de eventos gerados por um grupo especializado de usuários como (i) professores, (ii) revisores, (iii) pesquisadores e/ou (iv) tutores, o que torna a recomendação mais relevante ao partir de tais usuários (NORMADHI et al., 2019). Desta forma, a responsabilidade das recomendações possui um fator que se desloca do indivíduo para o coletivo (VESIN et al., 2013). Isto está diretamente relacionado com o segundo pilar.

Sistemas de recomendação exibem **inteligência coletiva**. Por exemplo, se um livro é recomendado ao aluno, esta recomendação não pode ser rastreada até o comportamento de nenhum usuário individualmente. Pelo contrário, é o comportamento da coletividade (ou no caso da filtragem colaborativa baseada no usuário) que deve ser responsável pela recomendação; pode-se argumentar, portanto, que estes sistemas exibem inteligência coletiva (MALONE; LAUBACHER; DELLAROCAS, 2009; HERLOCKER et al., 2017). Esta ideia corrobora a noção de cognição de grupo, particularmente, em pesquisas sobre aprendizagem colaborativa suportada por computador (TRAUSAN-MATU; STAHL; SARMIENTO, 2006). De acordo com esta visão, a saída de um grupo de aprendizagem colaborativa, por exemplo, suas avaliações ou artefatos construídos não podem ser rastreados, de maneira significativa ou completa, até os membros individuais do grupo, mas surgem por meio de interações complexas geradas entre os membros do grupo (VESIN et al., 2013). Pode-se dizer que estas propriedades emergentes de grupos também podem ser encontradas na maneira como os sistemas de recomendação operam.

Terceiro, os sistemas de recomendação são **baseados no controle do usuário**. Um livro que é sugerido por um sistema de recomendação difere de um livro que é uma parte obrigatória do currículo do curso. Desta forma, o aluno tem a opção de seguir a

recomendação ou não. Os sistemas de recomendação preservam a autonomia do aluno e não prescrevem ações a serem tomadas por uma pessoa. Eles geralmente oferecem suporte à pesquisa e recuperação de informações, ou seja, tarefas de natureza auto-dirigida, exploratória e frequentemente aberta. Neste sentido, tais sistemas atendem a epistemologias construtivistas modernas das ciências da aprendizagem, que também enfatizam a importância da aprendizagem auto-regulada ou de descoberta (VESIN et al., 2013; NORMADHI et al., 2019).

Quarto, os sistemas de recomendação **fornecem orientações**. O aluno não se depara com uma lista enorme de todas as publicações sobre o tópico de interesse, mas recebe uma lista filtrada dos títulos que são mais relevantes para sua pesquisa. Ao dar instruções e oferecer dicas que um usuário pode ou não considerar, os sistemas de recomendação são equivalentes a um guia. Fornecer orientação também é uma questão central nas ciências da aprendizagem, pois muita autonomia do aluno pode ser percebida como onerosa e pouca autonomia, como as restrições presentes nos modelos convencionais (VESIN et al., 2013; WILLINGHAM, 2019). A chave é encontrar um delicado equilíbrio entre autonomia e orientação, para que a orientação não se torne imaterial nem diretiva demais.

Por fim, em quinto lugar, os sistemas de recomendação são personalizados, ou seja, **sugerem itens adaptados** às necessidades, interesses e preferências de cada aluno. A noção de personalização também desempenha um papel importante nas ciências da aprendizagem: diferentes alunos não se beneficiam, no mesmo grau, de tipos uniformes de instrução, e existe um consenso geral de que o material instrucional deve ser adaptado ao conhecimento, às necessidades e às habilidades de cada alunos (WILLINGHAM; HUGHES; DOBOLYI, 2015; WILLINGHAM, 2019). Consequentemente, tecnologias de aprendizagem, como sistemas de tutoria inteligentes ou ambientes hipermídia adaptativos, adaptam as informações às necessidades e habilidades dos alunos. Os sistemas de recomendação baseiam-se na mesma ideia geral ao corresponder sua saída ao perfil historicamente desenvolvido do aluno.

Adicionalmente, Aprendizado Adaptativo em sistemas de *e-learning* possibilita benefícios para os demais agentes do ambiente educacional (GOMEDE et al., 2018; GAFFO et al., 2017). Um dos pontos favoráveis é a comunicação entre aluno e professor feita de forma recomendada, o que permite uma ação mais eficiente e eficaz para resolver eventuais questões relevantes. Da mesma maneira, as recomendações adaptadas permite com que o sistema consiga **gerenciar a heterogeneidade** dos alunos. Da mesma forma, reduz a sobrecarga de informação, o que pode ser um problema, principalmente em níveis iniciais (WILLINGHAM, 2019). De forma complementar, os principais elementos a serem recomendados pelos sistemas estão relacionados com objetos de aprendizagem no contexto de uma atividade específica. Por exemplo, ao assistir um vídeo sobre como escrever

uma rede neural artificial o sistema pode recomendar um artigo sobre como realizar a implementação em *Python*. Além disso, uma sequência de objetos de aprendizagem pode ser recomendada para manter o engajamento e ajustar o nível de dificuldade.

Desta forma, os princípios identificados nas **ciências de aprendizagem** – compartilhar a responsabilidade para com grupo, aproveitar a inteligência coletiva, habilitar o controle do usuário, fornecer guias e adaptarem-se às necessidades, habilidades e interesses – estão incorporados em muitas tecnologias de informação, sendo que os sistemas de recomendação personalizados combinam todos esses princípios (NORMADHI et al., 2019). Neste sentido, tais sistemas exibem características com potencial para alavancar os processos de aprendizado (WILLINGHAM; HUGHES; DOBOLYI, 2015; BERNARD et al., 2017; WILLINGHAM, 2019). No entanto, a utilização dos sistemas de recomendação aos contextos de aprendizagem de forma alguma implica que tais sistemas devam ser adaptados às peculiaridades e requisitos do ambiente educacional (GAFFO et al., 2017; GOMEDE et al., 2018).

Nesta perspectiva, um ponto central do Aprendizado Adaptativo em sistemas de *e-learning* reside no conceito **produtor/consumidor**. Este conceito distingue entre a função de destinatário, em que os alunos são confrontados com itens recomendados, e a função de produtor em que os alunos geram dados, que são a base para os cálculos do sistema. Para funcionar corretamente, em contextos educacionais, os sistemas de recomendação devem fornecer o tipo certo de informação para que o aprendizado das recomendações seja ativado (função do destinatário). Além disso, o funcionamento adequado dos sistemas de recomendação exige que o usuário gere dados nos quais as computações do sistema possam ser executadas (função de produtor). Além dessas adaptações, centradas no sistema, o segundo problema explorado para as funções de destinatário e produtor refere-se a adaptações sociais. O processamento da informação humana em geral e o aprendizado em particular podem ser caracterizados pela racionalidade limitada (CHANG; LIN; CHEN, 2019). A navegação e a seleção de itens em um sistema de recomendação (função de destinatário) e a classificação de itens (função de produtor) são influenciadas por vários fatores psicológicos e sociais que podem ser vinculados à racionalidade limitada. Por exemplo, nem sempre o aluno se atenta às informações das quais se aprende mais (TANG; MCCALLA, 2005); e nem sempre se contribui com informações, mesmo que uma comunidade inteira possa se beneficiar de tal atividade. Portanto, os sistemas de recomendação educacionais podem ser aprimorados com a introdução de adaptações sociais, que facilita a tendência de processamento de informações propícias ao aprendizado ou atenuam os vieses que são prejudiciais ao aprendizado.

Existem duas maneiras de definir o problema de recomendação em relação à sua saída (WANG; WU, 2011; WANG; WANG; YEUNG, 2015).

- Problema de previsão: espera-se que o sistema preveja as interações do aluno com objetos de aprendizagem que ainda não foram utilizados. Isto também pode ser visto como um **problema de conclusão de matriz**, porque é baseado em uma matriz incompleta de aluno/objeto.
- Problema de recomendação: espera-se que o sistema selecione k objetos. Esta é uma definição mais real do problema, porque é o que geralmente é feito com os objetos selecionados. Esse problema também é chamado de **problema de recomendação $top@k$** .

Em termos práticos, o objetivo do Aprendizado Adaptativo em sistema de *e-learning* é melhorar o resultado do aluno. No entanto, este não é um objetivo operacional. Existem vários objetivos operacionais comuns que devem ser tratados com a finalidade de melhorar a qualidade do Aprendizado Adaptativo (BARBIERI et al., 2017).

- Relevância: os objetos recomendados devem ser relevantes para o aluno.
- Novidade: é importante que os objetos recomendados não tenham sido vistos pelo aluno antes. Por exemplo, há muito pouco valor em recomendar o artigo mais popular de um determinado assunto, porque o aluno provavelmente já o viu.
- *Serendipity*: parece ser benéfico, se às vezes a recomendação é surpreendente para o aluno. Isto significa que não é apenas um novo objeto semelhante aos que o aluno já viu. Isto o ajuda a expandir um pouco seus horizontes além de aumentar a diversidade de aprendizagem.
- Diversidade: normalmente, a recomendação final é uma lista dos principais objetos exibidos para o aluno. Se todos os objetos forem muito semelhantes, o aluno poderá não gostar de nenhum deles. Uma lista mais diversificada aumenta as chances de, que pelo menos, um dos objetos seja interessante.

O aprendizado, especialmente o auto-regulado, é uma tarefa demorada que requer responsabilidade por parte do aluno. Isto pode ser auxiliado por um sistema de recomendação (NORMADHI et al., 2019). Um sistema de recomendação educacional visa tornar o aprendizado mais eficiente e eficaz. A **eficiência** descreve a maneira de atingir uma meta em termos de esforços, processos e tempo, já a **eficácia** diz respeito diretamente ao resultado alcançado. Em termos de aprendizado, uma maior eficiência significa otimizar o processo, economizando esforço e tempo para atingir o mesmo objetivo do curso. Uma eficácia mais alta significa alcançar uma meta mais alta, por exemplo, uma nota melhor no exame ou um conhecimento mais duradouro. Ambos podem ser aprimorados por meio do suporte à decisão dos sistemas de *e-learning*. Kraus (KRAUSS, 2016) apresenta os seguintes critérios para recomendações adequadas:

- Transparência: explica como o sistema funciona
- Verificação: permite que os alunos digam ao sistema o que está errado
- Confiabilidade: aumenta a confiança dos alunos no sistema
- Satisfação: aumenta a usabilidade e a resposta do sistema de recompensas do cérebro
- Persuasão: convence os alunos a experimentar, motivando a aprendizagem
- Eficiência: ajuda os alunos a tomar decisões mais rapidamente
- Eficácia: ajuda os alunos a tomar boas decisões

Portanto, o Aprendizado Adaptativo em um sistema de *e-learning* deve respeitar todos estes critérios. Contudo, a eficiência e a eficácia das recomendações dos objetos de aprendizagem são de particular interesse no aprendizado aprimorado por tecnologia, porque os dois critérios afetam diretamente os estilos de aprendizado e ensino (KRAUSS, 2016).

2.2 Comparativo

Nesta seção, os trabalhos relacionados à classificação de estilos de aprendizagem e recomendação de objetos são discutidos e comparados. Isto permite uma visão geral da pesquisa atual desta área e os fundamentos para comparações com os resultados propostos. Inicialmente, os trabalhos relacionados com a classificação, uma forma menos granular de Aprendizado Adaptativo, são revisados e comparados com a finalidade de obter uma linha mestre sobre o assunto e fornecer as bases para a comparação com a proposta desta tese. Por fim, os trabalhos relacionados com a recomendação de objetos de aprendizagem, uma forma mais granular de Aprendizado Adaptativo, foram comparados em dois grupos; os que lidam com questões computacionais e os que aplicam as questões computacionais para resolver problemas educacionais.

2.2.1 Classificação

Segundo Truong (TRUONG, 2016) e Normadhi (NORMADHI et al., 2019), os pesquisadores têm procurado mecanismos para detectar, automaticamente, os estilos de aprendizagem dos alunos com base em diferentes modelos. O processo de detecção automática do estilo de aprendizagem pode ser dividido em três subproblemas: (i) selecionar um modelo de aprendizado adequado, (ii) selecionar os descritores e os alvos para representar o comportamento *on-line* do aluno (em um MOOC) e (iii) selecionar o algoritmo e parâmetros adequados ao problema de previsão. Este procedimento é mostrado na Figura 2.1.

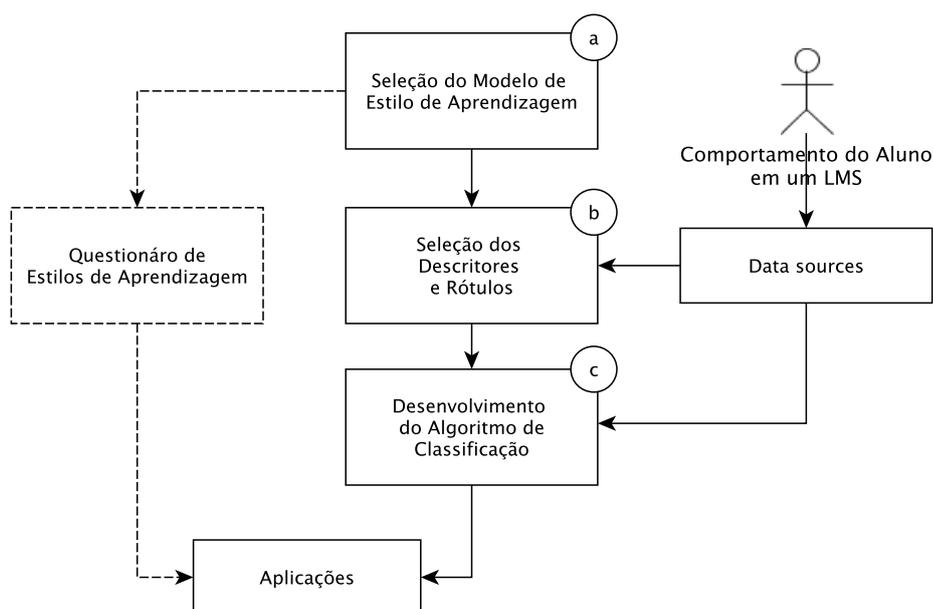


Figura 2.1 – Processo para classificação automática de estilos de aprendizagem. Os pontos estudados nesta tese foram os A, B e C. A linha tracejada mostra a detecção via questionário (GOMEDE; BARROS; MENDES, 2020).

Conforme mostrado na Figura 2.1, para classificar os estilos de aprendizagem dos alunos alguns pesquisadores se concentram no uso de algoritmos enquanto outros se concentram na aplicação desse modelo usando métodos tradicionais, como questionários (linha tracejada).

2.2.1.1 Seleção do Modelo de Estilo de Aprendizagem

Segundo Willingham (WILLINGHAM, 2019), qualquer teoria de estilo de aprendizagem deve prever que um método de ensino específico pode ser adequado para uma pessoa, mas não necessariamente para outra. Portanto, para otimizar a capacidade de aprendizado de um aluno, pode-se explorar estes diferentes métodos de aprendizado. Neste sentido, é imperativo compreender a diferença entre capacidade de aprendizagem e estilo de aprendizagem. **Capacidade de aprendizado** é a capacidade ou o sucesso em certos tipos de disciplinas (matemática, por exemplo). Em contraste com as habilidades, o **estilo de aprendizagem** é uma tendência ao raciocínio de uma maneira específica, por exemplo, aprendizagem sequencial ou holística. Apesar desta afirmação, há uma diferenciação entre a popularidade das abordagens de estilos de aprendizagem na educação e a falta de evidências confiáveis de sua utilidade (WILLINGHAM; HUGHES; DOBOLYI, 2015; WILLINGHAM, 2019). Conforme indicado por Pashler (PASHLER et al., 2008), ainda deve ser demonstrado se a caracterização dos estilos de aprendizagem dos alunos tem alguma utilidade razoável. No entanto, uma investigação de Kolb e Kolb (KOLB; KOLB, 2005) examina os avanços em andamento nas hipóteses e nas pesquisas sobre

aprendizagem experimental e explora como ela pode ajudar a melhorar a aprendizagem no ensino superior. Além disso, Kolb e Kolb presumiram que os estilos de aprendizagem se baseiam na pesquisa e observação clínica destes modelos e podem ser aplicados em todo o ambiente educacional por meio de um programa de desenvolvimento institucional.

Neste sentido, vários componentes dos estilos de aprendizagem foram pesquisados, conceitual e empiricamente (BERNARD et al., 2017). Além disso, inúmeras hipóteses e múltiplas taxonomias tentando descrever como as pessoas raciocinam e aprendem foram propostas, organizando os indivíduos em grupos distintos. Ainda, como indicado por Omar (BA-OMAR; PETROUNIAS; ANWAR, 2007), diferentes instrumentos de estilo de aprendizagem foram produzidos para fins de pesquisa e pedagogia. Segundo a teoria de Felder-Silverman (FSLSM) (FELDER; SILVERMAN et al., 1988), existem 4 dimensões que descrevem os estilos de aprendizagem: **processamento**, **percepção**, **entrada** e **entendimento**. De acordo com Truong (TRUONG, 2016) e Normadhi (NORMADHI et al., 2019), muitos pesquisadores desenvolveram modelos de detecção automática baseados no FSLSM, cujas 4 dimensões são diretamente derivadas de seus 4 objetivos. A dimensão de **processamento** caracteriza os alunos entre **ativos** e **reflexivos**, identificados pelo seu interesse em realizar atividades físicas ou teóricas. Estudantes ativos são os indivíduos que preferem trabalhar em grupo e realizar inúmeras atividades, enquanto estudantes reflexivos preferem trabalhar sozinhos e realizar alguns exercícios. Os alunos **sensitivos** e **intuitivos** são caracterizados pela dimensão da **percepção**. Os alunos sensitivos são aqueles que têm mais atenção e cuidado e normalmente alcançam seus objetivos com poucas tentativas, apresentando alta taxa de conclusão do exercício e atingindo alto desempenho nos exames. Por outro lado, os alunos intuitivos geralmente ficam entediados com os detalhes, mostram descuido e só alcançam seus objetivos com várias tentativas que apresentam baixa taxa de conclusão do exercício e atingem baixo desempenho nos exames. A dimensão de **entrada** reconhece os alunos por sua inclinação no conteúdo **visual** ou **verbal** e nos processos ao estudar e participar de atividades em grupo. Finalmente, a dimensão do **entendimento** decide se os alunos se inclinam para a metodologia **sequencial** ou **global** na obtenção de assuntos de estudo. Os estudantes sequenciais preferem avançar em direção ao estudo e às informações de maneira sequencial, semelhante a um roteiro, enquanto os estudantes globais preferem obter uma visão geral e depois mergulhar nos detalhes, tentando compreender pontos específicos e vincular estas informações a outros (BERNARD et al., 2017).

Portanto, para iniciar a determinação automática do estilo de aprendizagem, a etapa inicial é escolher um modelo razoável de estilos de aprendizagem. Atualmente, existem mais de 70 modelos propostos, alguns deles sobrepondo suas abordagens de seleção. Como apontado por Truong (TRUONG, 2016), estes modelos sofrem alguns problemas em termos de validade e confiabilidade, com a maioria deles apresentando desempenhos semelhantes. A partir disso, o modelo de estilo de aprendizagem de Felder-

Silverman (FSLSM) é frequentemente usado para identificação automática de estilos de aprendizagem. Em um trabalho pioneiro, relacionado a classificação automática de estilos de aprendizagem, Graf *et al.* (GRAF; KINSHUK; LIU, 2009) apresentam três razões para selecionar o FSLSM: (i) usa 4 dimensões, permitindo uma classificação mais detalhada; (ii) descreve a preferência de reunir, processar e armazenar informações; e (iii) trata cada dimensão como um tendência em vez de um tipo absoluto. Estas dimensões podem ser vistas como um *continuum*, com uma inclinação de aprendizado na extrema esquerda e outra na extrema direita, conforme Saryar (SARYAR *et al.*, 2019).

2.2.1.2 Seleção de Descritores e Variáveis Alvo

Existem três fontes principais de recursos que foram reconhecidas: arquivos de *log*, informações estáticas, como os históricos de usuário e outras fontes de personalização. As fontes potenciais de dados e os atributos correspondentes podem ser resumidos da seguinte forma:

- Arquivos de *log*: esta fonte coleta o comportamento dos usuários quando eles estão interagindo com o MOOC. É possível, neste caso, obter informações como o número de visitas e o tempo gasto em vários objetos de aprendizagem, como conteúdo, estrutura de tópicos, auto-avaliação, exercício, questionário, fórum, pergunta, navegação e exemplos (RASHEED; WAHID, 2019; GRAF; KINSHUK; LIU, 2009; EL-BISHOUTY *et al.*, 2019; LI; RAHMAN, 2018; FRANZONI *et al.*, 2008).
- Histórico do usuário e dados de pessoais: inclui informações estáticas ou lentas, como recursos pessoais (gênero, idade, etc.). Tais informações raramente são incorporadas na classificação, embora pesquisas anteriores tenham indicado que estas variáveis podem assumir um papel fundamental na determinação dos estilos de aprendizagem (TRUONG, 2016).
- Outras fontes de personalização: elas podem incorporar conhecimento prévio, capacidade inteligente, atributos cognitivos (capacidade de memória de trabalho, velocidade de processamento, habilidades de aprendizado e capacidade de raciocínio), objetivos do estudo, idioma e nível de motivação, que, em alguns casos, podem ser considerados próximos aos estilos de aprendizagem (TRUONG, 2016).

Independentemente dos vários descritores considerados, nenhuma pesquisa abordou como diferentes atributos contribuem para a previsão de estilos de aprendizagem. A descoberta de tais comparações pode desempenhar um papel importante na melhoria da eficiência e eficácia de diferentes previsões.

2.2.1.3 Desenvolvimento do Modelo de Classificação e Avaliação

Uma das estratégias mais populares usadas para classificação e avaliação é o algoritmo baseado em regras, no qual os pesquisadores interpretam diferentes estilos, de acordo com as hipóteses, em diferentes regras estatísticas. Este método é usado nas regras da Rede Bayesiana e Naïve Bayes. Além disso, outros algoritmos como a Rede Neural Artificial, Algoritmo Genético, Otimização de Colônias de Formigas, Otimização de Enxame de Partículas e Árvore de Decisão também podem ser aplicados para classificação. Entre estes algoritmos, o que realiza a precisão ideal é a **Rede Neural Artificial** (RNA) (BERNARD *et al.*, 2017). A maneira comum usada para avaliar o modelo é a divisão do conjunto de dados nos subconjuntos de treinamento e teste (RASCHKA, 2018).

2.2.1.4 Trabalhos Relacionados

Em um artigo de revisão, Truong (TRUONG, 2016) apresenta um estudo resumindo vários trabalhos em uma visão geral dos modelos utilizados na classificação de estilos de aprendizagem. Este artigo analisa 51 trabalhos dividindo a classificação do estilo de aprendizagem em três subproblemas. Segundo o autor, os modelos podem ser categorizados entre aqueles que podem mudar ao longo do tempo, aqueles que podem mudar ao longo de situações e aqueles que não podem mudar. Além disso, a utilização de estilos de aprendizagem fornece aos instrutores uma ferramenta para compreender seus alunos. Truong também mostra que há uma associação entre estilos de aprendizagem e escolhas de carreira. Com base nisso, podem ser desenvolvidas recomendações e orientações para apoiar o planejamento de carreira. O autor também divide os estudos entre aqueles que apenas classificam o estilo de aprendizagem e aqueles que fazem previsões baseadas em descritores fornecidos pelo comportamento do usuário. Este último é usado para personalização e recomendação em sistemas de *e-learning*.

Em outro trabalho de revisão, Normadhi *et al.* (NORMADHI *et al.*, 2019) afirmam que as técnicas usadas para reconhecer as características da personalidade podem ser divididas em três categorias: (i) questionários, (ii) detecção por computador e (iii) ambos. Estratégias de identificação baseadas em computador são usadas com mais frequência para melhorar a obtenção de traços de personalidade em um perfil de aluno, analisando a entrada implícita do usuário. Estas técnicas são consideradas mais precisas do que as técnicas de questionário, porque respondem rapidamente a mudanças nas características da personalidade do aluno, caso ocorram. As técnicas de reconhecimento baseadas em computador podem ser classificadas como aprendizado de máquina, aprendizado não-máquina e técnicas híbridas. Além disso, as técnicas de detecção baseadas em computador podem ser importantes para novos alunos, pois as informações são inicialmente insuficientes para construir os perfis de um aluno apropriadamente. Além disso, os autores afirmam que a maioria dos pesquisadores utiliza traços de personalidade na categoria domínio

da aprendizagem cognitiva (62,82%), no ambiente de aprendizagem adaptativa e na dimensão do modelo, frequentemente utilizados no modelo Felder-Silverman (FSLSM). Os autores também afirmam que os resultados das técnicas de identificação têm uma grande influência positiva em ambientes de aprendizado adaptativo. Por exemplo, explorar a avaliação observacional para ambientes adaptativos de *e-learning* é especialmente relevante. Pesquisas que conduzem experimentos para comparar a eficácia e o desempenho das técnicas de identificação também são altamente incentivadas. Finalmente, exames futuros devem explorar e investigar as forças e fraquezas dos traços de personalidade que são mapeados no objeto de aprendizagem e nos materiais selecionados.

Em um artigo original, Bernad *et al.* (BERNARD *et al.*, 2017) investigaram 4 algoritmos de inteligência computacional (rede neural artificial, algoritmo genético, otimização de colônias de formigas e otimização de enxame de partículas) para melhorar a precisão da detecção do estilo de aprendizagem. Como resultado, os autores alcançaram uma precisão média de 80% usando a Rede Neural Artificial. Os autores também apontaram os inconvenientes dos usos do questionário, como: (i) presume-se que os alunos estejam motivados a preencher o questionário, (ii) eles preencherão o formulário completamente (sem influência) e (iii) entenderão como preferem aprender. Adicionalmente, os autores utilizaram o FSLSM e descritores de comportamento relevantes de Graf *et al.* (GRAF; KINSHUK; LIU, 2009) Os autores vincularam estes descritores a estilos de aprendizagem, indicando que cada descritor está associado a um estilo de aprendizagem. Estes descritores são baseados em diferentes tipos de objetos de aprendizado, incluindo *outlines*, conteúdos, exemplos, autoavaliações, questionários, exercícios e fóruns. Os descritores consideram quanto tempo um aluno permaneceu em um determinado tipo de objeto de aprendizagem (por exemplo, *content_stay*) e com que frequência ele visita um determinado tipo de objeto de aprendizagem (por exemplo, *content_visit*). Além disso, as perguntas foram classificadas com base nos conceitos, se exigem detalhes ou uma visão geral do conhecimento, se incluem gráficos ou se apenas o texto é utilizado. Estas perguntas também tratam do desenvolvimento ou interpretação de soluções. Além disso, os autores apresentam métricas para avaliar os resultados. O desempenho das abordagens propostas é medido usando quatro métricas: (i) SIM (semelhança), (ii) ACC (precisão), (iii) LACC (menor precisão) e (iv)% de correspondência (porcentagem de alunos identificados com um nível razoável de precisão).

Em outro artigo original, Sheeba e Krishnan (SHEEBA; KRISHNAN, 2019) propõem uma maneira de lidar com a classificação automática do estilo de aprendizagem dos alunos com base em seu comportamento de aprendizagem. Esta abordagem é baseada em um classificador de árvore de decisão para o desenvolvimento de regras significativas necessárias para distinguir com precisão os estilos de aprendizagem. Esta abordagem foi experimentada em 100 alunos em um curso *on-line* criado no Sistema de Gerenciamento de Aprendizado Moodle, ou *Learning Management System* (LMS). Neste experimento,

os autores alcançaram a precisão média de 87% nas dimensões de processo, percepção e entrada. Os autores também apresentaram dois métodos utilizados para o reconhecimento automático de estilos de aprendizagem; (i) abordagem orientada a dados e (ii) baseada em literatura. A **abordagem orientada a dados** usa dados de amostra para criar um classificador que imita um instrumento de estilo de aprendizagem. Esta abordagem usa principalmente o algoritmo de classificação Inteligência Artificial (IA), que toma o modelo do aluno como entrada e retorna aos alunos as preferências do estilo de aprendizagem como saída. A **abordagem baseada na literatura** utiliza regras simples para calcular estilos de aprendizagem a partir da quantidade de dicas correspondentes. Eles usam um conjunto de dados de arquivos de *log* da web contendo todos os comportamentos que o aluno executa no *Moodle LMS*. Estes *logs* são criados automaticamente quando os alunos usam o sistema. Eles registram todas as atividades por meio de bate-papos, fóruns, questionários, exercícios, tarefas, entrega de exames e frequência de acesso aos materiais do curso.

Assim, este trabalho visa contribuir com os trabalhos analisados propondo métodos e procedimentos para superar as limitações encontradas. Identificou-se, primeiro, que os descritores, de estudos anteriores, estão relacionados com **dimensões específicas** no modelo computacional. No entanto, o modelo psicológico, FLSM (FELDER; SILVERMAN *et al.*, 1988), não segue a mesma abordagem. Por exemplo, uma visita do aluno ao resumo do curso, ativando o descritor *outline_visit*, pode ser interpretada como uma característica exclusiva da dimensão da percepção (sensitivo/intuitivo) (PASHLER *et al.*, 2008; BERNARD *et al.*, 2017). Portanto, investigou-se a interferência de todos os descritores nas quatro dimensões usando uma técnica de classificação múltipla. Segundo, a estratégia para rotular o conjunto de dados é vaga. Os *logs* prevêm que o MOOC não possui o rótulo do estilo de aprendizagem, apenas o comportamento. Os autores não fornecem um método claro para **rotular** o conjunto de dados usado no modelo de treinamento (BERNARD *et al.*, 2017). Além disso, eles não apresentam problemas comuns relacionados ao conjunto de dados, como o conjunto de dados desbalanceados (RASCHKA, 2018). Terceiro, o **contexto** do conjunto de dados não é descrito de maneira compreensível. Por exemplo, os autores apresentam as informações do nível dos alunos (estudantes de graduação), mas não indicam a idade média, tipo de curso, duração do curso, frequência e resultados (aprovado/reprovado). Além disso, relacionados às técnicas de inteligência computacional, os autores não fornecem uma estratégia explícita para superar problemas de *overfitting*, estratégia para atingir os parâmetros ideais (como número de camadas ocultas em uma rede neural artificial) e estratégia para treinar e testar os modelos de construção. Finalmente, há uma falta de **métricas de desempenho**, como *f-score*, *recall*, precisão, sensibilidade e outras. Isso prejudica a possibilidade de comparação com trabalhos relacionados (atuais e futuros) e não apresenta análises diferentes dos resultados dos testes.

2.2.2 Recomendação

Grande parte dos trabalhos publicados sobre sistemas de recomendação foi dedicada a questões de implementação técnica, modelagem matemática e métricas de desempenho. No entanto, existe uma crescente conscientização de que questões não técnicas devem ser levadas em consideração para melhorar os sistemas de recomendação personalizados, principalmente se estes sistemas forem aplicados em ambientes não tradicionais, como a educação. Conseqüentemente, alguns autores começaram a teorizar sobre a possibilidade dos sistemas de recomendação considerando questões educacionais (TANG; MCCALLA, 2005; DRACHSLER et al., 2009; MANOUSELIS et al., 2011; KRAUSS, 2016).

Um dos primeiros trabalhos com a aplicação de um *Auto Encoder* em sistemas de recomendação foi proposto por Wang (WANG; WANG; YEUNG, 2015). Este tipo de arquitetura de rede neural artificial *feedforward* é considerado semi-supervisionado e não exige um conjunto de dados rotulados. Trabalha com o método de filtragem colaborativa baseado em eventos de *feedback* dos usuários para cada item (XUE et al., 2017) e apresenta bons resultados comparado a outros métodos de filtragem colaborativa (ZHANG et al., 2019). As camadas escondidas, responsáveis pela codificação e decodificação, criam o espaço latente que representa a fatoração da matriz usuário/item de entrada. O número de camadas pode variar de acordo com a matriz esparsa de entrada (STRUB; GAUDEL; MARY, 2016), entretanto, a tendência da função de perda (RMSE) é estabilizar com um comportamento logarítmico (SEDHAIN et al., 2015; XUE et al., 2017; DONG et al., 2017; HARUNA et al., 2017). Algumas das possíveis limitações sobre este tipo de técnica estão relacionadas com a possibilidade de interpretação; uma vez que o processo de treinamento é estocástico, o resultado dos pesos da rede é de difícil explicação. Requisitos de dados também é outra limitação, pois para o treinamento da rede é necessário um conjunto suficiente de observações. Por fim, a utilização de hiperparâmetros para o ajuste do modelo (ZHANG et al., 2019).

Uma arquitetura de *Auto Encoder* é utilizada para preencher os valores faltantes da matriz esparsa na camada de saída e, desta forma, permitir a lista ordenada de itens para um usuário (ZHANG et al., 2019). Uma das métricas utilizadas para utilizadas para a predição das recomendações é a *Recall@M*, onde M é a quantidade dos itens a serem recomendados (LI; SHE, 2017). Outro aspecto importante são os tipos de *feedback* utilizados. Algumas abordagens utilizam o *feedback* explícito e, outras, implícito e explícito (CHEN; PENG, 2018; CHAE; KIM; LEE, 2019). Um ponto importante a ser destacado deve-se ao fato dos autores não deixarem explícito as estratégias utilizadas para divisão do conjunto de dados para treinamento e testes, principalmente relacionadas com técnicas de amostragem que podem gerar modelos tendenciosos (ZHAO et al., 2018). Este ponto é importante para melhorar o resultado do modelo gerado. A comparação entre os trabalhos pode ser vista na Tabela 2.1.

Tabela 2.1 – Comparação entre os trabalhos relacionados com relação as questões técnicas

Autor(es)	Pré	Arquitetura	Divisão	Treinamento	Feedback	Dataset	Métricas
(WANG; WANG; YEUNG, 2015)	•	Auto Encoder		Backpropagation	Explícito	CiteULike	Recall@M
(SEDHAIN et al., 2015)	•	Auto Encoder	90/10	Backpropagation	Implícito	MovieLens	RMSE
(STRUB; GAUDEL; MARY, 2016)	◦	Auto Encoder	90/10	Backpropagation	Implícito	MovieLens	RMSE
(XUE et al., 2017)		Matrix Factorization	LOOCV	Backpropagation	Implícito	MovieLens	NDCG
(LI; SHE, 2017)	◦	Auto Encoder		Backpropagation	Implícito	CiteULike	Recall@M
(DONG et al., 2017)	•	Auto Encoder	90/10	Backpropagation	Explícito	Book-Crossing	RMSE
(HARUNA et al., 2017)	•	Algorithm		Backpropagation	Explícito	Paper	MAP
(ZHAO et al., 2018)		Q-learning			Ambos		RMSE
(CHEN; PENG, 2018)		Matrix co-factorization		Backpropagation	Ambod	MovieLens	NDGC
(CHAE; KIM; LEE, 2019)		Auto Encoder		Backpropagation		Watcha	MSE
Este	•	Deep Auto Encoder	80/20	Backpropagation	Implícito	Cursos	RMSE

* Para o pré-processamento • indica que houve criação da matriz esparsa e normalização. Já ◦ indica que houve somente a criação da matriz esparsa. Ausência de marcação indica que os autores não informaram no artigo. As divisões são, exceto quando indicada, *holdout*.

A utilização deste tipo de técnica no contexto educacional pode ser vista no trabalho de Bobadilla (BOBADILLA; SERRADILLA; HERNANDO, 2009). Alguns dos objetivos citados pelos autores refere-se a redução de erros, apresentando uma acurácia mais alta possível; além disso, evitar a super especialização, encontrar bons itens de recomendação e sua credibilidade, precisão, *recall*, entre outros. Os autores apresentam também uma série de itens a serem recomendados, desde objetos de aprendizagem, como livros, textos, testes, assuntos até professores (BOBADILLA; SERRADILLA; HERNANDO, 2009). Isso fornece uma visão mais abrangente do sistema de recomendação no contexto educacional. Os objetos de aprendizagem podem ser padronizados, e dessa forma gerenciados de uma melhor maneira, por meio da utilização do modelo conhecido como *Sharable Content Object Reference Model* (SCORM) (WANG; WU, 2011; MEDIO et al., 2020). Estes objetos podem ser utilizados para um aprendizado de vários níveis educacionais, fornecendo um sistema de recomendação que pode ir do fundamental até a pós-graduação e pesquisa (PROTASIEWICZ et al., 2016). Além disso, este sistema pode ser adaptado para ser ubíquo, via dispositivos móveis, o que permite o acesso dos interessados no ambiente educacional em qualquer local a qualquer momento (WANG; WU, 2011).

Já o trabalho de Nafea (NAFEA; SIEWE; HE, 2019) apresenta um sistema de recomendação de objetos de aprendizagem baseados no modelo SCORM. Este trabalho utiliza um modelo de perfil de usuário baseado no *Felder-Silverman Learning Style Model* (FSLSM). Este modelo, apesar de várias citações na literatura, necessita de evidências científicas para comprovar sua eficácia (WILLINGHAM; HUGHES; DOBOLYI, 2015). No trabalho de Medio (MEDIO et al., 2020) pode-se observar a utilização de repositórios de objetos de aprendizagem para gerar a recomendação de objetos para os alunos. Este trabalho apresenta a avaliação em um cenário com dados reais com um aumento no engajamento dos alunos devido às recomendações feitas pelo sistema. Esta técnica apresenta melhores resultados, em termos de recomendação, se comparada com *Principal Component Analysis* (PCA) e *Singular Value Decomposition* (SVD) (BARBIERI et al., 2017). Um comparativo entre os trabalhos que abordam questões educacionais e/ou pedagógicas é mostrado na Tabela 2.2.

Tabela 2.2 – Comparação entre os trabalhos relacionados com relação as questões educacionais e/ou pedagógicas

Autor(es)	Contribuição
(BOBADILLA; SERRADILLA; HERNANDO, 2009)	Usuários com maior resultado em testes tem maior peso na recomendações
(WANG; WU, 2011)	Utilização de informação de contexto (baseado em redes e sensores) para recomendar assuntos para objetivos individuais
(KRAUSS, 2016)	Recomendação de uma lista de objetos de aprendizagem orientada ao tempo
(PROTASIEWICZ et al., 2016)	Recomendação de revisores para artigos científicos
(BARBIERI et al., 2017)	Trabalho mais similar ao desta tese com uma arquitetura <i>Auto Encoder</i> para recomendação de objetos de aprendizagem
(NAFEA; SIEWE; HE, 2019)	Recomendação de objetos de aprendizagem utilizando os estilos do Felder-Silverman
(MEDIO et al., 2020)	Recomendação de cursos baseados em uma API para o LMS Moodle
Este	Recomendação de cursos utilizando <i>Auto Encoder</i> para prever e gerar uma lista de objetos de aprendizagem baseados no padrão SCORM

Por exemplo, um sistema de recomendação de objetos de aprendizagem precisa aprender as preferências de seus usuários para poder, por meio de seu aprendizado, executar a tarefa de recomendar tais objetos corretamente. Segundo (CHANG; LIN; CHEN, 2019), para possuir um problema de aprendizagem bem formulado, é necessário identificar três características: a tarefa, a medida de desempenho e a experiência de treinamento. Usando o exemplo acima, é possível identificar estas três características da seguinte maneira:

- Tarefa T : recomendar objetos de aprendizagem
- Medida de desempenho P : erro entre a classificação real e prevista
- Experiência de treinamento E : aprendendo com as classificações dadas pelos usuários

O objetivo da tarefa de aprendizado é determinar uma hipótese que procure generalizar o problema em todas as instâncias do conjunto de treinamento. A definição geral do problema de recomendação pode ser dada como:

Definição 1 *Um algoritmo aprende pela experiência E em respeito a um conjunto de tarefas T e uma medida de desempenho P , se a sua medida de desempenho nas tarefas T , medido por P , melhora com a experiência E .*

No que trata das análises de recomendação, este trabalho visa contribuir com os anteriores fornecendo uma análise dos eventos (*feedback* implícitos e explícitos) que possam ser utilizados para a construção da matriz esparsa e com a análise dos objetos de aprendizagem que possam ser utilizados. Além disso, pretende-se também apresentar quatro formas de personalização, baseada em arquiteturas de redes neurais artificiais conhecidas como *Deep Auto Encoder*. Por fim, pretende-se também fornecer experimentos com métricas de comparação e estratégias para otimização de parâmetros que permitam trabalhos posteriores evoluírem esta pesquisa.

2.3 Conclusão

O *e-learning* é o processo de fornecer instrução *on-line* na Internet aos alunos, para que estes possam estudar e aprender em qualquer lugar e dispositivo de computação (computador pessoal, telefone celular, tablet, etc.), mídia eletrônica, informações, tecnologias e plataformas da Internet, como *Learning Management Systems* (LMS) (GARRIDO; MORALES; SERINA, 2016). Este tipo de pesquisa evoluiu desde a década de 60 até os dias atuais. Esta evolução e os termos que a acompanham podem ser vistos na Figura 2.2.

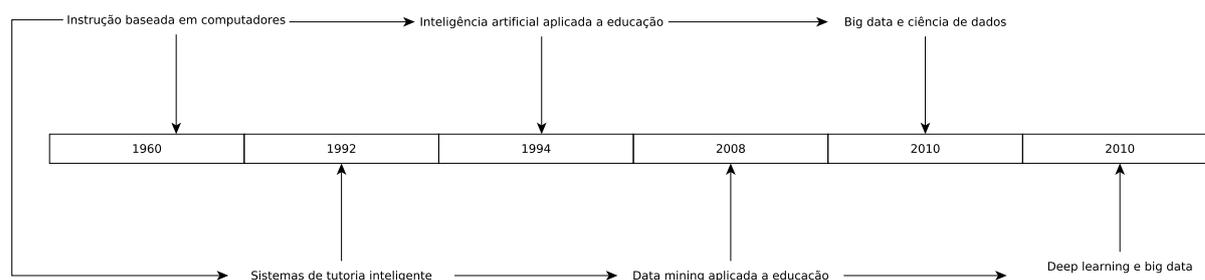


Figura 2.2 – Resumo da evolução da computação aplicada ao contexto educacional. Pode-se notar que, a partir dos anos 60, as primeiras soluções começaram a ser propostas com o objeto de ser um ferramental a disposição dos interessados no ambiente.

Pode-se notar na Figura 2.2 que as técnicas foram sendo aplicadas de acordo com o estado da arte no momento de sua publicação e que os conceitos educacionais foram sendo melhorados de acordo com a tecnologia utilizada. Embora a quantidade de assuntos abordados com algoritmos de inteligência computacional seja variada (NORMADHI et al., 2019), pode-se resumi-los em estratégias de Aprendizado Adaptativo em ambiente de *e-learning*. Estas estratégias visam, no geral, adaptar a resposta do sistema às preferências dos alunos com o objetivo de melhorar sua taxa de aprendizagem. Portanto, pode-se concluir que os trabalhos são baseados no princípio fundamental que pode ser descrito como a **utilização de inteligência computacional para Aprendizado Adaptativo em sistemas de *e-learning***.

3 Conceitos Fundamentais

Este capítulo apresenta os conceitos sobre arquiteturas de redes neurais artificiais que foram utilizados para a construção dos experimentos. Estes fundamentos são utilizados como base pelos Capítulos 4 e 5, tanto para o projeto, a elaboração e a execução dos experimentos, quanto para a discussão e as análises dos resultados. Além disso, este capítulo fornece os fundamentos sobre sistemas de *e-learning* e demais assuntos relacionados, tais como objetos de aprendizagem, repositórios, modelos de referências e estilos de aprendizagem.

3.1 Redes Neurais Artificiais

Uma rede neural é um processador distribuído paralelo, composto de processamento de unidades com habilidade para armazenar conhecimento experimental e disponibilizá-lo para uso (HEIDARI et al., 2020). Assemelha-se ao cérebro humano em dois aspectos: (i) o conhecimento é adquirido pela rede a partir de seu ambiente por meio de um processo de aprendizado; (ii) os pontos fortes da conexão interneurônio, conhecidos como pesos sinápticos, são usados para armazenar o conhecimento necessário. A Figura 3.1 apresenta a arquitetura mais simples das redes neurais artificiais, o *perceptron*¹.

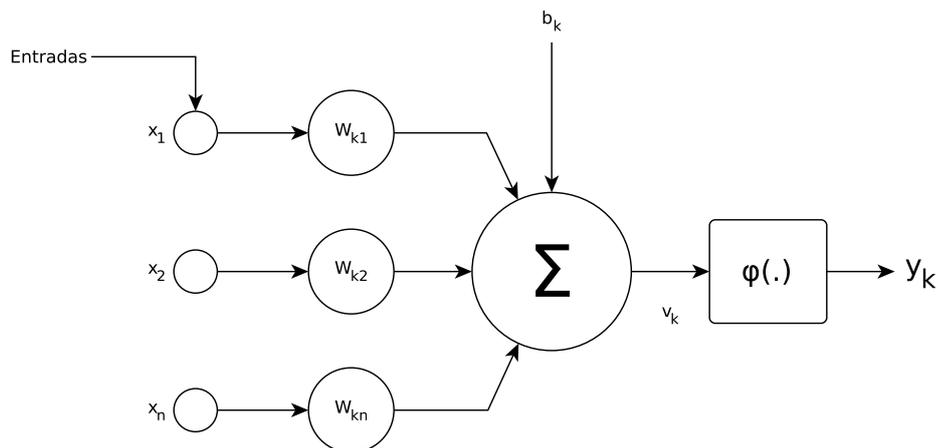


Figura 3.1 – Exemplo de uma rede neural *perceptron*. Normalmente, a faixa de amplitude normalizada da entrada/saída de um neurônio é escrita como o intervalo unitário fechado $[0, 1]$ ou, alternativamente, $[-1, 1]$ (HEIDARI et al., 2020).

¹ O *perceptron* é um tipo de rede neural artificial inventada em 1958 por Frank Rosenblatt no *Cornell Aeronautical Laboratory*. Ele pode ser visto como o tipo mais simples de rede neural *feedforward*: um classificador linear.

Em termos matemáticos, pode-se descrever o neurônio k , representado na Figura 3.1, pelas Equações 3.1 e 3.2:

$$u_k = \sum_{j=1}^n w_{kj} \times x_j \quad (3.1)$$

$$y_k = \psi(u_k + b_k) \quad (3.2)$$

onde x_1, x_2, \dots, x_n são os sinais de entrada; $w_{k1}, w_{k2}, \dots, w_{kn}$ são os respectivos pesos sinápticos do neurônio k ; u_k (não mostrado na Figura 3.1) é a saída do combinador linear devido à entrada de sinais; b_k é o viés; $\psi(\cdot)$ é a função de ativação; e y_k é o sinal de saída do neurônio. Um exemplo do algoritmo pode ser visto no Pseudocódigo 3.1 (HEIDARI et al., 2020).

Algorithm 3.1 Algoritmo de Treinamento da Rede Neural Artificial *Perceptron*

Input: Training data S , learning rate η , initialization σ , epoch θ

Output: Model parameters $\Theta = (w_0, \mathbf{w}, \mathbf{V})$

$w_d \leftarrow w_+ y x_d$, for all $s = 1 \dots S$

$b \leftarrow o$

for $i = 1 \dots epoch$ **do**

foreach $(x, y) \in S$ **do**

$a \leftarrow \sum_{s=1}^S w_d x_d + b$ **if** $ya \leq o$ **then**

$w_d \leftarrow w_+ y x_d$, for all $s = 1 \dots S$ $b \leftarrow b + y$

end

end

end

A rede *perceptron* é o tipo de arquitetura mais simples (HEIDARI et al., 2020). O treinamento é feito iterando pelo número de épocas especificado como hiperparâmetro e ajustando os pesos de cada sinapse e do *bias*. Este tipo de rede pode ser utilizado para problemas de classificação com separação linear.

3.2 Multi Layer Perceptron

O *perceptron* de múltiplas camadas é uma das arquiteturas de redes neurais artificiais mais utilizadas (HEIDARI et al., 2020). Na maioria das vezes, os sinais são transmitidos dentro da rede em uma direção: da entrada à saída. Como não há *loop*, a saída de cada neurônio não afeta o próprio neurônio. Camadas que não estão diretamente conectadas ao ambiente são chamados ocultas. Na literatura, há controvérsia quanto à primeira camada (a camada de entrada) sendo considerada autônoma (própria) camada da rede, pois sua única função é transmitir os sinais de entrada para os estratos superiores, sem nenhum processamento nas entradas. Desta forma, serão consideradas apenas as camadas

que consistem em neurônios independentes. Há também redes de retroalimentação, que podem transmitir impulsos em ambas as direções, devido às conexões na rede. Estes tipos de redes são úteis para a resolução de um vasto número de problemas; entretanto, podem ser extremamente complexos. São dinâmicos, mudando sua condição o tempo todo, até a rede atingir um estado de equilíbrio e a busca de uma nova matriz de peso ocorre a cada entrada mudança. A introdução de várias camadas foi determinada pela necessidade de aumentar a **complexidade das regiões de decisão** (HEIDARI et al., 2020). Como mostrado no exemplo anterior, um *perceptron*, com uma única camada e uma entrada gera regiões de decisão sob a forma sinapses. Adicionando outra camada, cada neurônio atua como um *perceptron* padrão para as saídas dos neurônios na camada anterior, assim a saída da rede pode estimar regiões de decisão convexas, resultante da interseção dos semi planos gerados pelos neurônios. A Figura 3.2 mostra um exemplo de arquitetura de uma rede neural artificial *Multi Layer Perceptron* (MLP).

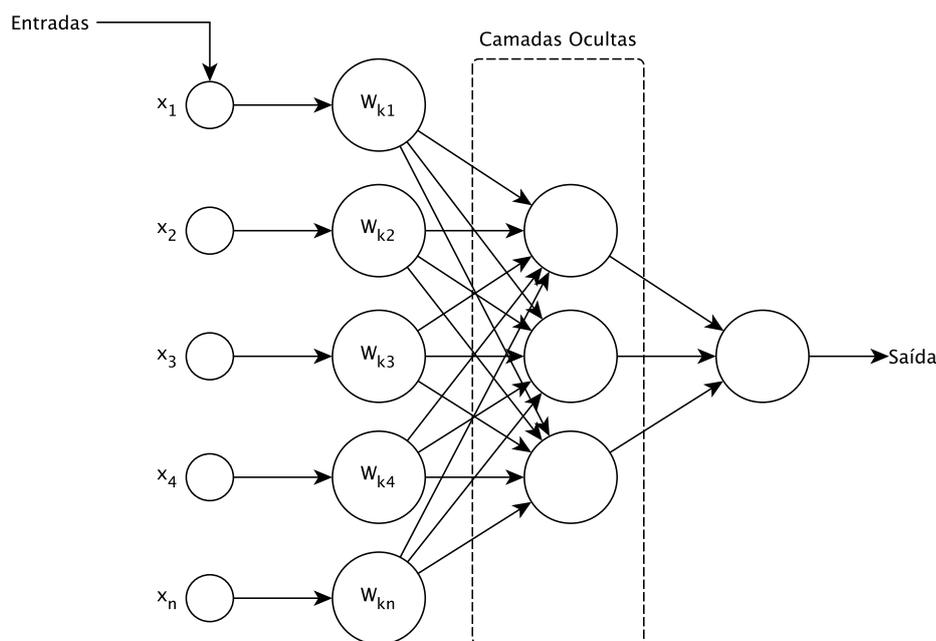


Figura 3.2 – Exemplo de uma rede neural de múltiplas camadas (*multi layer perceptron*). Normalmente, a faixa de amplitude normalizada da entrada/saída de um neurônio é escrita como o intervalo unitário fechado $[0, 1]$ ou, alternativamente, $[-1, 1]$. Além disso, a quantidade de neurônios e camadas são hiperparâmetros da rede que devem ser otimizados para o problema em questão (HEIDARI et al., 2020).

Em relação à função de ativação de neurônios, verificou-se que as redes multicamadas não fornecem um aumento no poder de computação em comparação com redes com uma única camada, se as funções de ativação forem lineares, porque um conjunto de funções lineares também é uma função linear. O poder do *perceptron* multicamada vem precisamente a partir de **funções de ativação não lineares**. Praticamente qualquer

função não linear pode ser usada para esta finalidade, exceto para funções polinomiais. Atualmente, as funções mais usadas são as sigmóides monopolar (ou logístico).

Pode-se notar que as funções sigmóides agem de forma linear para pequenos valores absolutos do argumento de entrada assumindo o papel de limiar para altos valores. Foi demonstrado (KHAN; HAYAT; PORIKLI, 2019) que uma rede com uma camada oculta (possivelmente com infinitas) é capaz de aproximar qualquer função contínua. Para alguns tipos de modelos, redes recorrentes (ou seja, *feedforward*), em que algumas interconexões e *loops*, também são usados. Nas redes *feedforward*, as unidades geralmente são organizadas em níveis (camadas), mas outras topologias podem ser usadas.

3.3 Deep Auto Encoder

Os *Deep Auto Encoders* (DEA) (também chamados de *Auto Associators*) são um dos tipos de arquitetura de redes neurais artificiais de aprendizagem profunda que reduz o número de dimensões de um conjunto de dados em um espaço latente. Uma vez que os dados são codificados por meio do algoritmo, estes são decodificados na camada de saída. O treinamento da rede é finalizado quando os dados de entrada são compatíveis com os dados de saída. Algumas das aplicações de *Deep Auto Encoders* envolvem a compressão de dados, redução de ruídos, sistemas de recomendação e seleção de características (que permite reduzir a dimensionalidade dos dados similares às técnicas de *Principal Component Analysis* (PCA), *Singular Value Decomposition* (SVD) e *t-Distributed Stochastic Neighbor Embedding* (t-SNE)) (BALDI, 2012; WANG; WANG; YEUNG, 2015; SEDHAIN et al., 2015; WU et al., 2016; STRUB; GAUDEL; MARY, 2016; LI; SHE, 2017; BARBIERI et al., 2017; KUCHAIEV; GINSBURG, 2017; ORDWAY-WEST; PARVEEN; HENSLEE, 2018; KHAN; HAYAT; PORIKLI, 2019).

Este tipo de arquitetura de rede é considerado como semi supervisionado, pois não existe a necessidade de rotular os dados para o treinamento. Apesar disso, o treinamento é feito comparando a camada de saída com a camada de entrada. Por exemplo, em uma rede com entrada $E = \{x_1, x_2, \dots, x_n\}$ a codificação e decodificação visa reconstruir esta entrada com o menor nível de erro possível. Desta forma, valores faltantes podem ser aproximados. Existem alguns autores que classificam essa arquitetura de rede como semi supervisionado (KHAN; HAYAT; PORIKLI, 2019). Maiores detalhes sobre os tipos de treinamentos de redes neurais artificiais, como divisão do conjunto de dados, amostragem, algoritmo de treinamento, algoritmo de otimização, técnicas e hiperparametrização, normalização, entre outros, podem ser vistos na Seção 3.7. A Figura 3.3 mostra um exemplo de arquitetura de redes neurais artificiais conhecidas como *Deep Auto Encoder*. Este é um modelo canônico e não utiliza nenhuma classe específica de arquitetura.

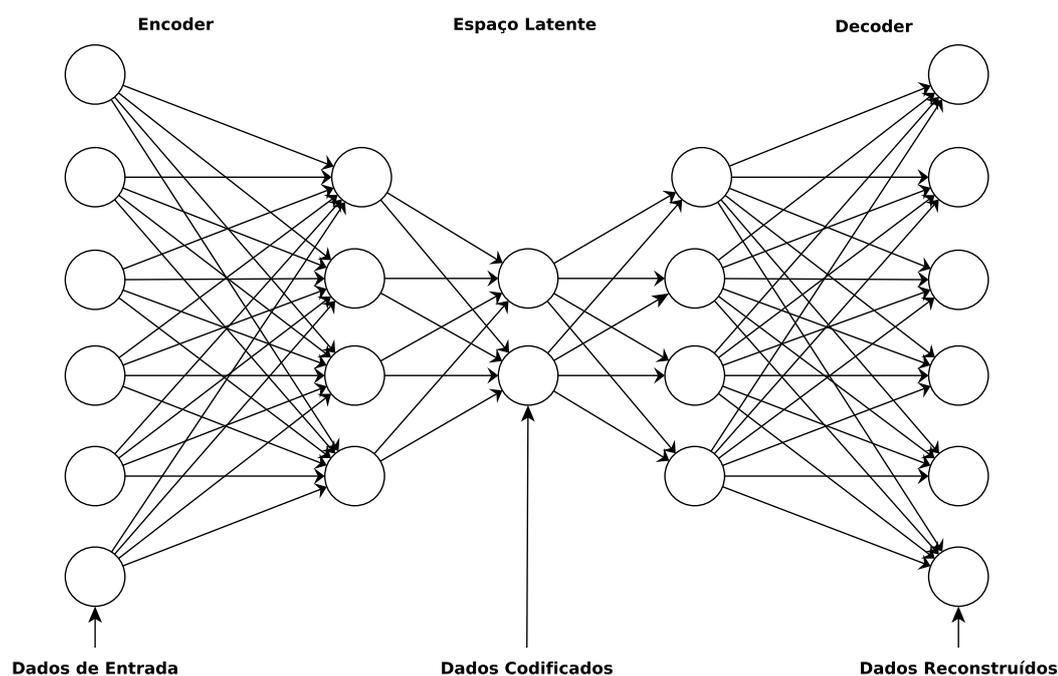


Figura 3.3 – Um *Deep Auto Encoder* (DAE) é uma categoria de algoritmos que são formados por duas partes, um *encoder* e um *decoder*. O *encoder* tem como função comprimir a informação da entrada em um espaço-latente diferente. Pode ser representado como um função de x , $f(x) = h$.

Um DAE é uma arquitetura *feedforward* similar a uma arquitetura *perceptron* de múltiplas (MLP) camadas. A diferença entra a DAE e a MLP é que o objetivo do AE é reconstruir a entrada de dados enquanto a MLP é predizer saídas alvo. O número de neurônios na camada de entrada é idêntico ao número de neurônios na camada de saída. No processo de codificação, a DAE converte um vetor de entrada x em uma representação escondida h usando uma matriz de pesos \mathbf{w} , então, no processo de decodificação a DAE mapeia h de volta para a entrada original x' com outra matriz de peso \mathbf{w}' . Em teoria, \mathbf{w}' deveria ser a transposta de \mathbf{w} .

3.4 Backpropagation

O algoritmo de retropropagação (*backpropagation*) foi introduzido na década de 1970, mas sua importância não foi totalmente apreciada até um famoso artigo de 1986 de David Rumelhart, Geoffrey Hinton e Ronald Williams. Este artigo descreve várias redes neurais em que a retropropagação funciona muito mais rápida do que as abordagens anteriores de aprendizagem, possibilitando o uso de redes neurais para resolver problemas que antes eram insolúveis. Atualmente, o algoritmo de retropropagação é utilizado para o aprendizado em redes neurais. A Figura 3.4 mostra um exemplo de arquitetura de redes neurais artificiais que utilizam o algoritmo de retropropagação (*backpropagation*),

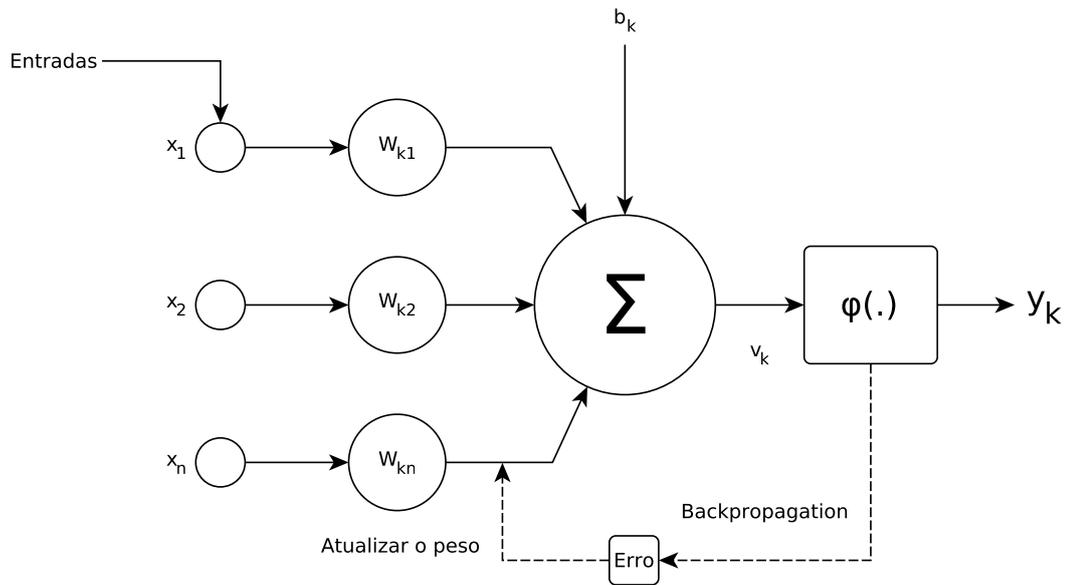


Figura 3.4 – A retropropagação é um algoritmo amplamente utilizado no treinamento de redes neurais artificiais *feedforward* para aprendizado supervisionado ou semi-supervisionado.

Como exemplo, pode-se considerar um *perceptron* multicamada com uma camada de entrada de m_0 neurônios, duas camadas ocultas e um único neurônio de saída, como mostrado na Figura 3.1. Os elementos do vetor de peso w são ordenados por camada (a partir da primeira camada oculta) e depois por neurônios e, em seguida, pelo número de sinapses dentro do neurônio (HEIDARI et al., 2020). Seja w_{ji} o peso sináptico do neurônio i ao neurônio j na camada $l = 1, 2, \dots, n$. Para $l = 1$, respondendo à primeira camada oculta, o índice i refere-se a um nó de origem e não a um neurônio. Para $l = 3$, correspondente à camada de saída exibida na Figura 3.1, tem-se j_3 . Deseja-se avaliar as todas as derivadas da função $F(w, x)$ em relação a todos os elementos do vetor de peso w para um vetor de entrada especificado $x = [x_1, x_2, \dots, x_m]^T$. Em resumo, o algoritmo de retropropagação é responsável por atualizar todos os pesos de todas as sinapses, entre os neurônios, em todas as camadas, ocultas ou não. Para isso, o algoritmo *backpropagation* precisa saber qual valor atualizar; isso é fornecido pelo **algoritmo de decida estocástica do gradiente** (BOTTOU, 2010; HEIDARI et al., 2020; FIORESI; CHAUDHARI; SOATTO, 2020) descrito na Seção 3.5. Um exemplo do algoritmo de retropropagação (*backpropagation*), com as fases *forward* e *backward*, é representado pelo Pseudocódigo 3.2 (FIORESI; CHAUDHARI; SOATTO, 2020).

Onde o vetor D representa o conjunto de dados de entradas, ou seja, as amostras que serão utilizadas para o treinamento da rede. Por sua vez, o vetor L representa o conjunto de camadas, ocultas ou não, da rede neural artificial. O algoritmo de retropropagação deve percorrer ambos os conjuntos para a atualização dos pesos de todas as sinapses entre todos

Algorithm 3.2 Algoritmo de Treinamento *Backpropagation*

```

for  $d \in D$  do
  Forwards Pass
  A partir da camada de entrada, use o algoritmo 3.1 por meio da Equação 3.1 para
  avançar através da rede, computando os pesos dos neurônios em cada camada
  Backwards Pass
  Calcule as derivadas da função de erro em relação à camada de saída
  for  $l \in L$  do
    Calcule as derivadas da função de erro em relação às entradas dos neurônios
    da camada superior
    Calcule as derivadas da função de erro com relação aos pesos entre a camada
    externa e a camada abaixo
    Calcule as derivadas da função de erro com respeito para as atividades da
    camada abaixo
  end for
  Atualize os pesos
end for

```

os neurônios (BOTTOU, 2010; HEIDARI et al., 2020; FIORESI; CHAUDHARI; SOATTO, 2020). A retropropagação do valor a ser atualizado é fornecida, para cada sinapse, via taxa calculada pelo algoritmo de descida do gradiente estocástico. Este algoritmo identifica o valor de atualização dos pesos que reflete na saída do modelo, visando minimizar a diferença entre as saídas esperadas e as previstas.

3.5 Descida do Gradiente Estocástico

A descida do gradiente é um **algoritmo de otimização** iterativo de primeira ordem para encontrar o mínimo local de uma função. Para isto, é necessário executar etapas proporcionais ao negativo do gradiente (ou gradiente aproximado) da função no ponto atual. Se, em vez disso, ao avançar de forma positiva, o gradiente se aproxima de um máximo local desta função, o procedimento é então conhecido como subida de gradiente. A descida do gradiente foi proposta originalmente por (CAUCHY, 1847 apud FIORESI; CHAUDHARI; SOATTO, 2020). A intuição subjacente a descida do gradiente estocástico pode ser ilustrada por um cenário hipotético. Uma pessoa está presa nas montanhas e está tentando descer, ou seja, tentando encontrar o mínimo global. Há nevoeiro de modo que a visibilidade é baixa. Portanto, o caminho descendo a montanha não é visível; assim, ela deve usar as informações locais para encontrar o mínimo. Sendo assim, pode usar o método de descida de gradiente, que envolve observar a inclinação da colina na sua posição atual e prosseguir na direção com a descida mais íngreme. Usando este método, ela acabaria descendo a montanha ou possivelmente ficaria presa em algum local (ou seja, o mínimo local). No entanto, suponha também que a inclinação da colina não é imediatamente óbvia com uma simples observação, mas exige um instrumento sofisticado para medir.

Existe um custo associado ao tempo para medir a inclinação da colina com o instrumento; portanto deve-se minimizar o uso. A dificuldade, então, é escolher a frequência com que deve-se medir a inclinação. Um exemplo do algoritmo pode ser visto no Pseudocódigo 3.3 (FIORESI; CHAUDHARI; SOATTO, 2020).

Algorithm 3.3 Descida do Gradiente Estocástico

Input: Dados de treinamento S , hiperparâmetros λ , aprendizagem η , inicialização σ

Output: Parâmetros do Modelo $\Theta = (w_0, \mathbf{w}, \mathbf{V})$

```

 $w_0 \leftarrow 0$ ;  $\mathbf{w} \leftarrow (0, \dots, 0)$ ;  $\mathbf{V} \sim \mathcal{N}(0, \sigma)$  repeat
  for  $(x, y) \in S$  do
     $w_0 \leftarrow w_0 - \eta(\frac{\partial}{\partial w_0} l(y(\mathbf{x} | \Theta), y) + 2\lambda^0 w_0)$  for  $i \in \{1, \dots, p\} \wedge x_i \neq 0$  do
       $w_i \leftarrow w_i - \eta(\frac{\partial}{\partial w_i} l(y(x | \Theta), y) + 2\lambda_\pi^w w_i)$  for  $f \in \{1, \dots, k\}$  do
         $v_{i,f} \leftarrow v_{i,f} - \eta(\frac{\partial}{\partial v_{i,f}} l(y(x | \Theta), y) + 2\lambda_{f,\pi(i)}^v v_{i,f})$ 
      end
    end
  end
until critério de parada não atingido;
  
```

Nesta analogia, a pessoa representa o algoritmo, o caminho percorrido a montanha e a utilização dos equipamentos os parâmetros que o algoritmo irá explorar. A inclinação da colina representa a inclinação da superfície de erro naquele ponto. O instrumento usado para medir a inclinação é a diferenciação (a inclinação da superfície do erro pode ser calculada tomando a derivada da função de erro quadrado neste ponto). A direção que ela escolhe para viajar alinha-se com o gradiente da superfície do erro naquele ponto. A quantidade de tempo que ela viaja antes de realizar outra medição é a taxa de aprendizado do algoritmo. Começando no ponto mais alto, da-se o primeiro passo na direção especificada pelo gradiente negativo. Em seguida, recalcula-se o gradiente negativo (passando nas coordenadas do nosso novo ponto) e da-se outro passo na direção especificada. Continua-se este processo iterativamente até chegar ao ponto mínimo do gráfico (mínimo global). O tamanho destas etapas é chamado de taxa de aprendizado. Com uma alta taxa de aprendizado, pode-se ir mais rápido, mas corre-se o risco de ultrapassar o ponto mais baixo, pois a inclinação está mudando constantemente. Com uma taxa de aprendizado muito baixa, pode-se avançar com confiança na direção do gradiente negativo, pois recalcula-se com tanta frequência. Uma baixa taxa de aprendizado é mais precisa, mas o cálculo do gradiente é demorado.

3.6 Matriz Esparsa

Em computação científica, uma **matriz esparsa** é uma matriz na qual a maioria dos elementos é zero ou desconhecido. Por outro lado, se a maioria dos elementos for diferente de zero ou conhecidos, a matriz será considerada uma **matriz densa**. O

número de elementos com valor zero dividido pelo número total de elementos (por exemplo, $m \times n$ para uma matriz $m \times n$) é chamado de **esparsidade da matriz** (que é igual a 1 menos a densidade da matriz). Usando estas definições, uma matriz será esparsa quando sua dispersão for maior que 0,5 (ZUO et al., 2016; DONG et al., 2017; LI; SHE, 2017; CHEN; PENG, 2018; SILVA; JUNIOR; CALOBA, 2018). Conceitualmente, a esparsidade corresponde a sistemas com poucas interações aos pares m/n . O conceito de esparsidade é útil em áreas combinatórias e de aplicação, como a teoria de redes, que possuem baixa densidade de dados ou conexões significativas; ou sistemas de recomendação com a interação usuário/item armazenada em uma matriz. Um exemplo de algébrico de matriz esparsa pode ser visto na Equação 3.3.

$$\begin{bmatrix} A_{-n,0}^{-n} & A_{-n+1,-1}^{-n} & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ A_{-n,1}^{-n+1} & A_{-n+1,0}^{-n+1} & A_{-n+2,-1}^{-n+1} & \ddots & & & & \vdots \\ 0 & A_{-n+1,1}^{-n+2} & A_{-n+2,0}^{-n+2} & A_{-n+3,-1}^{-n+2} & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & A_{n-3,1}^{n-2} & A_{n-2,0}^{n-2} & A_{n-1,-1}^{n-2} & 0 \\ \vdots & & & & \ddots & A_{n-2,1}^{n-1} & A_{n-1,0}^{n-1} & A_{n,-1}^{n-1} \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & A_{n-1,1}^n & A_{n,0}^n \end{bmatrix} \quad (3.3)$$

Ao armazenar e manipular matrizes esparsas em um computador é benéfico, e geralmente necessário, o uso de algoritmos e estruturas de dados especializados que tirem vantagem da estrutura esparsa da matriz. Algoritmos e estrutura de dados especializadas foram criadas para matrizes esparsas, pois tais matrizes são comuns no campo de aprendizado de máquina. As operações que utilizam estruturas e algoritmos padrão de matriz densa são lentas e ineficientes quando aplicadas a grandes matrizes esparsas, pois o processamento e a memória são desperdiçados nos valores desconhecidos. Os dados esparsos são, por natureza, mais facilmente compactados e, portanto, requerem significativamente menos armazenamento (SILVA; JUNIOR; CALOBA, 2018).

3.7 Tipos de Aprendizagem de Máquina

Os algoritmos de aprendizado de máquina são organizados em taxonomia, com base no resultado desejado do algoritmo. Os tipos de algoritmos comuns incluem 6 estratégias de aprendizagem (SATHYA; ABRAHAM, 2013):

Supervised Learning Nesta estratégia, o algoritmo gera uma função que mapeia entradas às saídas desejadas. Uma utilização comum de aprendizado supervisionado é o

problema de classificação: o algoritmo deve aprender (para aproximar o comportamento) uma função que mapeia um vetor, por exemplo, os objetos acessados por um aluno, em uma das classes disponíveis, observando vários exemplos de entrada para o treinamento. Portanto, existe a necessidade de dados rotulados ².

Unsupervised Learning Nesta estratégia, o algoritmo procura encontrar similaridades entre as observações disponíveis para criar grupos que permitam identificar comportamentos parecidos. Não existe a necessidade de dados rotulados.

Semi-supervised Learning Nesta estratégia, a combinação de dados rotulados e não rotulados são feitas para a construção da função que representa o comportamento procurado. Também é comum a utilização dos dados de entrada como rótulos como, por exemplo, no caso dos *Auto Encoders*.

Reinforcement Learning Nesta estratégia, o algoritmo aprende uma política de como agir, dada uma observação do mundo. Toda ação tem algum impacto no meio ambiente e o ambiente fornece um *feedback* que orienta o algoritmo no aprendizado. Este tipo de aprendizagem é comum em sistemas com agentes autônomos.

Transduction Learning Esta estratégia é semelhante à aprendizagem supervisionada, mas não constrói explicitamente uma função; em vez disso, tenta prever novos resultados com base em entradas de treinamento, saídas e novas entradas (GAMMERMAN; VOVK; VAPNIK, 1998). Em outras palavras, as entradas geram saídas que, por sua vez, são utilizadas para prever novos casos (VOLKWYN et al., 2019).

Learning to Learn ou Meta Learning Desta forma, o algoritmo aprende a como aprender e nenhum modelo precisa ser especificado. Esta última estratégia ainda não é comumente utilizada e também é conhecida como *Meta Learning*.

A análise computacional e de desempenho dos algoritmos de aprendizado de máquina é um ramo da estatística conhecida como **teoria da aprendizagem computacional**. O aprendizado de máquina é sobre como projetar algoritmos que permitem ao computador aprender. Do ponto de vista computacional, a ideia de aprender não envolve necessariamente consciência, mas é uma questão de encontrar estatísticas, regularidades ou outros padrões nos dados. Assim, muitos algoritmos de aprendizado de máquina pouco se assemelham a como os humanos podem abordar uma tarefa de aprendizado. No entanto, algoritmos de aprendizado podem dar uma ideia da dificuldade relativa de se aprender em diferentes ambientes (SATHYA; ABRAHAM, 2013).

² Rótulos em um conjunto de dados significa que uma determinada observação foi associada a uma classe. Por exemplo, um aluno foi classificado, na dimensão de percepção, como sensitivo. Tais rótulos não estão disponíveis na maioria dos casos.

3.8 Divisão do Conjunto de Dados

Em uma situação ideal não haveria a necessidade de dividir um conjunto de dados. Entretanto, o pré-processamento de tal conjunto de dados tende a ser uma tarefa custosa, tanto do ponto de vista computacional quanto de pesquisa, o que implica em uma estratégia de divisão para utilização em várias tarefas, como treinamentos e testes (RASCHKA, 2018).

3.8.1 Estratégias para Divisão

Em cenários onde somente um conjunto de dados está disponível, algumas formas de divisão podem ser aplicadas para criar um subconjunto de dados para as tarefas de treinamento, validação e testes. A Figura 3.5 mostra a possibilidade de divisão em 2 e 3 subconjuntos.

Conjunto de Dados Original		
Treinamento		Testes
Treinamento	Validação	Testes

Figura 3.5 – Duas estratégias para a divisão do conjunto de dados. A divisão em treinamento e testes e a divisão em treinamento, validação e testes. A estratégia mais comum é a de treinamento e testes (em destaque) (RASCHKA, 2018).

Dessa forma, ao utilizar uma das duas estratégias, pode-se obter os seguintes subconjuntos de dados com as finalidades:

Treinamento Este subconjunto é utilizado para o treinamento do modelo, no caso uma rede neural artificial. Ele é a base para a geração dos pesos da rede e fornece as observações necessárias para obter um modelo de maior acurácia.

Validação Este subconjunto é utilizado para a avaliação do modelo gerado. Por exemplo, em um classificador binário pode-se utilizar este subconjunto para avaliar a taxa de predição do modelo para um novo conjunto de dados.

Teste Este subconjunto é utilizado para, durante a etapa de treinamento, prever o desempenho do modelo. Esta predição é feita utilizando diversas métricas de desempenho. Detalhes sobre estas métricas são fornecidas nas Seções 4.2.6 e 4.1.5.

Pode-se utilizar estes subconjuntos em 5 estratégias de divisão. A estratégia nula (Estratégia 1) é quando todos os dados disponíveis são usados para todas as tarefas.

Treinamento, validação e testes com os mesmos dados geralmente leva ao ajuste excessivo do modelo e uma estimativa otimista da precisão preditiva, conhecido como *overfitting*. O erro estimado no mesmo conjunto que o modelo foi treinado é conhecido como erro de re-substituição. A estratégia motivada pela utilização de novos dados (Estratégia 2) utiliza um subconjunto para treinamento e o segundo subconjunto, contendo o primeiro subconjunto e os novos dados, para a avaliação. Mesclar novos dados coletados com os dados antigos reduz a independência da avaliação do modelo, o que pode levar a uma estimativa otimista do desempenho. A estratégia mais usada consiste em dividir os dados em dois subconjuntos, um subconjunto de treinamento e um subconjunto de testes. O subconjunto de treinamento (também chamado de conjunto de estimativa) é usado para estimar os parâmetros do modelo e também para validação do modelo. O conjunto de testes é então usado para avaliar a previsão desempenho (Estratégia 3). Outra estratégia (Estratégia 4) que divide os dados em dois conjuntos usa um subconjunto para aprendizado e o segundo para avaliar seu desempenho preditivo. O uso de um subconjunto independente para cada tarefa é geralmente recomendado. Por fim, a Estratégia 5 divide os dados disponíveis em três subconjuntos distintos para treinamento, teste e validação (RASCHKA, 2018; XU; GOODACRE, 2018). A Tabela 3.1 apresenta um resumo das estratégias considerando os três subconjuntos possíveis.

Tabela 3.1 – Resumo das estratégias para divisão do conjunto de dados em subconjuntos

Estratégia	Treinamento	Validação	Teste
1	Todos os Dados	Todos os Dados	Todos os Dados
2	Subconjunto 1	Todos os Dados	Todos os Dados
3	Subconjunto 1	Subconjunto 1	Subconjunto 2
4	Subconjunto 1	Subconjunto 2	Subconjunto 2
5	Subconjunto 1	Subconjunto 2	Subconjunto 3

Com estas estratégias duas novas questões surgem: (i) o tamanho de cada subconjunto e (ii) quais observações estarão dentro de cada um. Ambas questões são a finalidade dos algoritmos de divisão. Muitos algoritmos de divisão de dados foram propostos. A qualidade e complexidade dos algoritmos diferem e nenhuma abordagem é superior, em geral (RASCHKA, 2018). Alguns algoritmos comumente usados são descritos na Seção 3.8.2 a seguir.

3.8.2 Algoritmos para Divisão

O método de *Holdout* é o mais simples utilizando um conjunto de dados original e dividindo-o, aleatoriamente, em dois conjuntos. Uma prática comum é usar 1/3 para testar e 2/3 para treinar ou metade/metade. Geralmente o desempenho do modelo aumenta com o aumento da quantidade de observações no subconjunto de testes. Em outras palavras, os dois subconjuntos podem ter distribuições diferentes. Além disso, se um conjunto de dados não for grande o suficiente, e geralmente não é, o método de divisão

é ineficiente no uso de dados. Por exemplo, em um problema de classificação uma ou mais classes podem estar ausentes em um dos subconjuntos, o que leva a uma estimativa ruim do modelo, bem como à sua avaliação. Para lidar com isso, algumas versões avançadas usam a técnica conhecida como estratificação. Amostragem estratificada é uma amostragem probabilística, em que um conjunto de dados original é dividido em grupos não sobrepostos, chamados estratos, e as instâncias são selecionados de cada estrato proporcionalmente à probabilidade apropriada. Tal método garante que cada classe seja representada com a mesma frequência nos dois subconjuntos. Mas ainda não impede o surgimento do viés nos conjuntos treinamento e de teste. Para melhor confiabilidade da estimativa de erro, os métodos são repetidos e a precisão resultante é calculada como uma média em todas as iterações, o que pode reduzir o viés. O método *Repeated Holdout* também é conhecido como validação cruzada de Monte Carlo, conjuntos de subamostragem ou de avaliação repetida (RASCHKA, 2018; XU; GOODACRE, 2018).

O método de reamostragem mais popular é a validação cruzada (*Cross-Validation*). Na validação cruzada *k-fold*, o conjunto de dados original é dividido em k subconjuntos *k-fold* disjuntos do mesmo tamanho, em que k é um parâmetro do método. Em cada uma das k iterações, um subconjunto é usado para avaliação e os restantes $k - 1$ subconjuntos para o aprendizado de modelo, como mostrado na Figura 3.6. Como no método de repetição, a precisão é a média de todas as iterações. Assim como o método de *Holdout*, a validação cruzada sofre um viés pessimista, quando k é pequeno. Aumentar a quantidade de *fold* reduz o viés, mas aumenta a variância da estimativa (RASCHKA, 2018). As experiências mostraram que bons resultados em domínios diferentes têm o método de validação cruzada com $k = 10$ (XU; GOODACRE, 2018), mas em geral o hiperparâmetro k deve ser ajustado ao problema. A validação cruzada *k-fold* é muito semelhante ao método de validação repetida com a vantagem de que todas as observações do conjunto de dados original são usadas para treinar e testar o modelo. Este método pode ser visto, com maiores detalhes de funcionamento, na Figura 3.6.

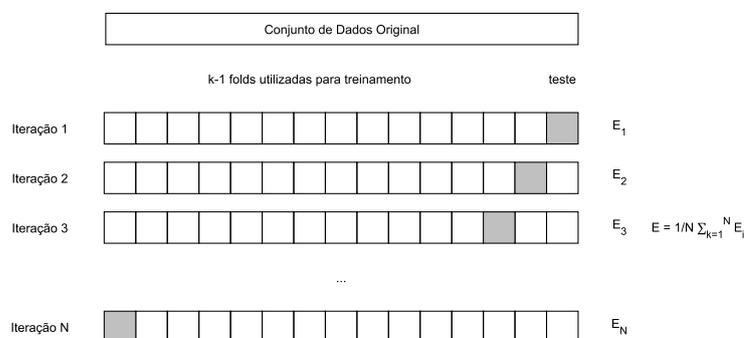


Figura 3.6 – O método *Cross-Validation* divide o conjunto de dados em k amostras (*folds*). Cada uma delas contém um número de observações. A cada iteração $k - 1$ amostras são selecionadas para o treinamento e a restante para o teste (RASCHKA, 2018).

A validação cruzada de uso único, ou ***Leave-one-out Cross-Validation*** (LOOCV) é o caso especial da validação cruzada k -fold em que $k = (n - 1)$, em que n é o tamanho do conjunto de dados original. Todos os conjuntos de teste sempre contêm apenas uma observação. Este método faz o melhor uso dos dados e não envolve nenhuma subamostragem aleatória. De acordo com isso, o LOOCV fornece estimativas quase imparciais do desempenho do modelo, mas geralmente com grande variabilidade. No entanto, este método é computacionalmente caro, o que geralmente o torna inaplicável dependendo do volume de dados de entrada (RASCHKA, 2018). A Figura 3.7 apresenta um exemplo deste método.

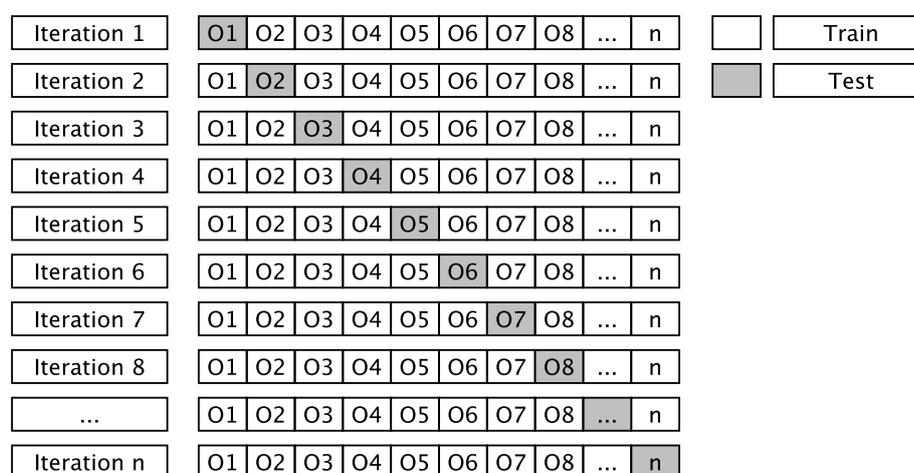


Figura 3.7 – O método *Leave-one-out Cross-Validation* (LOOCV) é um caso extremo do método k -fold cross validation. Uma amostra é usada para teste e as demais para treinamento (RASCHKA, 2018).

A ideia principal do método ***Bootstrap*** é descrita como segue. Dado um conjunto de dados S , de tamanho n , gere amostras de *Bootstrap* B por amostragem uniforme (com substituição), para n instâncias do conjunto de dados. Deve-se notar que a amostragem com substituição permite selecionar a mesma observação mais de uma vez. Após a nova amostragem, deve-se estimar os hiperparâmetros do modelo em cada amostra e, em seguida, estime o desempenho de previsão do modelo no conjunto de dados original. O erro de previsão geral é dado pela média destas estimativas B . O processo para construção e utilização dos subconjuntos de dados via *Bootstrap* é mostrado esquematicamente na Figura 3.8.

A abordagem mais conhecida e mais usada é o ***Bootstrap .632***. O número 0.632 significa que a fração esperada de instâncias distintas do conjunto de dados original apareceu no conjunto de treinamento. Cada instância tem uma probabilidade de $1/n$ para serem selecionadas de n instâncias ($(1 - 1/n)$ para não ser selecionado). Dá a probabilidade de $(1 - 1/n)^n \approx e - 1 \approx 0,368$ para não serem selecionadas após n amostras. Em outras palavras, esperamos que 63,2% instâncias do conjunto de dados original serão selecionados

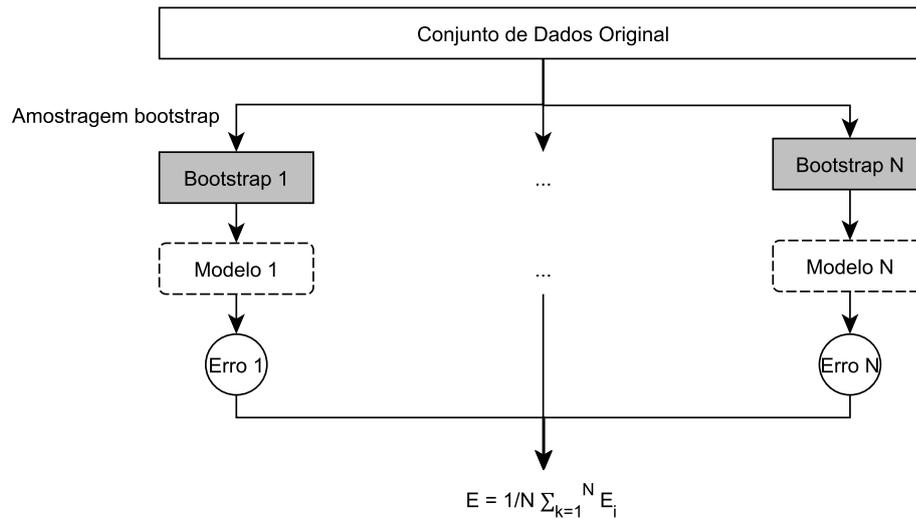


Figura 3.8 – O método de *Bootstrap* cria várias amostragens que sejam o mais próximas possíveis do conjunto de dados original. Estas amostras são utilizadas para criação de modelos diferentes. Estes modelos são comparados para verificar o resultado de cada um deles. Este tipo de método é conhecido como *ensembler* (RASCHKA, 2018).

para treinamento e 36,8% das instâncias restantes serão usadas para teste. O 0.632 estimativa de inicialização é definida como (Equação 3.4)

$$Acc(T) = \frac{1}{B} \sum_{i=1}^B (.632) \times Acc(B_i)_{B'_i} + .368 \times Acc(B_i)_T \quad (3.4)$$

onde $Acc(B_i)_{B'_i}$ é a precisão do modelo criado com a amostra de inicialização B_i como o conjunto de treinamento e aplicado ao conjunto de teste B'_i e $Acc(B_i)_T$ é a precisão do mesmo modelo aplicado ao conjunto de dados original. Os resultados mostram que as estimativas de *Bootstrap* de 0.632 geralmente apresentam baixa variabilidade, mas com um grande viés em comparação com a validação cruzada que fornecem aproximadamente estimativas, mas com alta variabilidade. Também é relatado que o *Bootstrap* 0.632 funciona melhor para pequenos conjuntos de dados (XU; GOODACRE, 2018).

O algoritmo de **Kennard-Stone (CADEX)** é usado para dividir os conjuntos de dados em dois subconjuntos A e B que cobrem aproximadamente a mesma região do espaço fatorial definido pelo conjunto de dados original. Em vez de medir a cobertura por um critério explícito, o algoritmo segue duas diretrizes. A primeira define que nenhuma observação do conjunto A deve estar muito longe de qualquer observação do conjunto B ; e a segunda aponta que a cobertura deve começar no limite do espaço fatorial. As observações são escolhidas sequencialmente e o objetivo é selecionar as observações, em cada iteração, para obter observações distribuídas uniformemente pelo espaço definido pelo conjunto de dados original (XU; GOODACRE, 2018). O algoritmo funciona da seguinte

maneira. Seja P o subconjunto de observações já selecionadas e seja Q o conjunto de dados igual a T no início. Define-se $Dist(p, q)$ como a distância da observação $p \in P$ a observação $q \in Q$ e $\Delta q(P)$ será a distância mínima da observação q sobre o conjunto de observações selecionadas em P . Isto pode ser visto na Equação 3.5.

$$\Delta_v^2(k) = \min_{i \in Q} \{D_{1v}^2, D_{2v}^2, \dots, D_{kv}^2\} \quad (3.5)$$

O algoritmo começa com a adição de duas instâncias mais distantes de Q a P (não é necessário selecionar as observações mais distantes, mas de acordo com a ideia cobertura, geralmente escolhe-se os dois casos mais distantes). Em cada iteração, o algoritmo seleciona uma instância das observações restantes no conjunto Q usando o critério (Equação 3.6)

$$\Delta_{k+1}^2 = \max_{i \notin Q} \{\Delta_v^2(k)\} \quad (3.6)$$

Em outras palavras, para cada instância restante no conjunto de dados Q , encontre as menores distâncias para instâncias já selecionadas em P e escolha aquela com a distância máxima entre estas distâncias menores. O processo é repetido até que objetos suficientes sejam selecionados. Primeira iteração do algoritmo é mostrada na Figura 3.11 (a) e na Figura 3.11 (b) é o resultado final com a área coberta por cada conjunto. Como o algoritmo usa distâncias, é sensível às métricas usadas e eventuais *outliers*. Para fins de classificação, os subconjuntos devem ser selecionados nas classes individuais. A versão aprimorada do CADEX denominada **DUPLEX**.

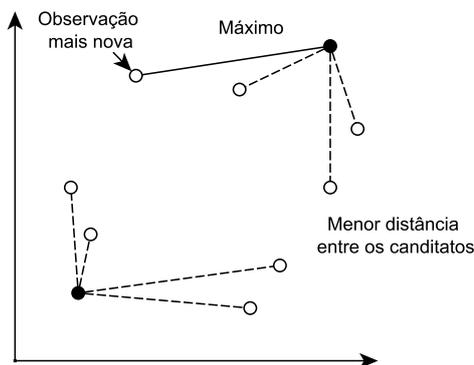


Figura 3.9 – Primeira iteração

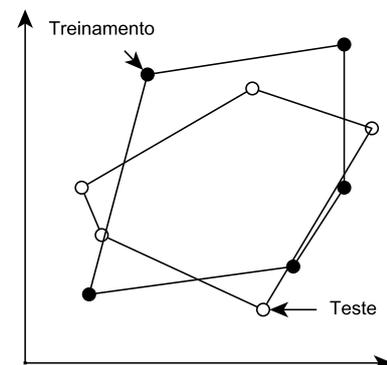


Figura 3.10 – Espaço de cobertura

Figura 3.11 – Divisão do conjunto de dados em dois subconjuntos que cobrem aproximadamente a mesma região do espaço fatorial definido pelo conjunto de dados original (RASCHKA, 2018).

Outros métodos podem ser considerados quando leva-se em consideração o seguinte pressuposto. Suponha que dois conjuntos P e Q formados pela divisão do conjunto de dados original S sejam tão semelhantes quanto possível quando a soma das distâncias de todos os pares (uma instância do par é de P e a outra de Q) são minimizados. Formalmente, (Equação 3.7)

$$d^* = \arg \min_d \sum_{(p,q) \in S} dist(p,q) \quad (3.7)$$

Encontrar a divisão ideal para os dois conjuntos é computacionalmente muito caro. Para isto, duas abordagens heurísticas foram propostas. A primeira é um método baseado na regra do vizinho mais próximo. Este método divide os conjuntos de dados originais em dois ou mais, encontrando o mais próximo instância (vizinho mais próximo) da instância escolhida aleatoriamente e colocando cada instância em um subconjunto diferente. A segunda heurística encontra o par mais próximo de instâncias em S e coloque uma instância em P e a segunda instância em Q . Isto é repetido até que o conjunto T esteja vazio. O resultado destes algoritmos são dois subconjuntos separados do conjunto de dados original. Um resumo com estes algoritmos, contendo os métodos de divisão, a categoria a qual pertencem, suas vantagens e desvantagens, é apresentado na Tabela 3.2.

Tabela 3.2 – Resumo com os algoritmos para divisão de dados

Algoritmo	Categoria	Vantagens	Desvantagens
<i>Holdout</i>	<i>Folds</i>	Simples	Determinar o tamanho da <i>fold</i>
<i>Repeated Holdout</i>	<i>Folds</i>	Simples	Determinar o tamanho da <i>fold</i>
<i>Cross-Validation</i>	<i>Folds</i>	Utiliza o conjunto de dados todo	Tamanho da <i>fold</i> e custo computacional
<i>Leave-one-out</i>	<i>Folds</i>	Todos usados para treinamento e testes	Custo computacional
<i>Bootstrap</i>	Amostras	Comparação de vários modelos	Custo computacional
<i>Bootstrap .632</i>	Amostras	Comparação de vários modelos	Custo computacional
CADEX/DUPLEX	Amostras	Mesmo espaço de dados fatorado	Custo computacional
<i>Random Sampling</i>	Amostras	Simples	Viés
<i>Convenience Sampling</i>	Amostras	Simples	Viés
<i>Stratified Sampling</i>	Amostras	Simples	Viés
<i>Stratified random sampling</i>	Amostras	Simples	Viés

Nesta pesquisa utilizou-se o *Leave-One-Out Cross Validation* para a classificação e o *Holdout Simple Random Sampling without Repetition* para as arquiteturas de recomendação. As escolhas foram feitas considerando-se a quantidade de dados disponíveis, balanceamento dos dados, custo computacional, complexidade do modelo e resultado da hiperparametrização.

3.9 Objetos de Aprendizagem

Um objeto de aprendizagem é uma coleção de itens de conteúdo, itens de prática e itens de avaliação que são combinados com base em um único objetivo de aprendizado. O termo é creditado a Wayne Hodgins e data de um grupo de trabalho em

1994. O conceito abrangido por objetos de aprendizagem é conhecido por vários outros termos, incluindo; objetos de conteúdo, blocos, objetos educacionais, objetos de informação, objetos inteligentes, objetos de conhecimento, componentes de aprendizado, reutilizáveis objetos de informação, objetos de aprendizado reutilizáveis, componentes de treinamento e unidades de aprendizado. A Figura 3.12 mostra uma organização de OA em um sistema de *e-learning*.

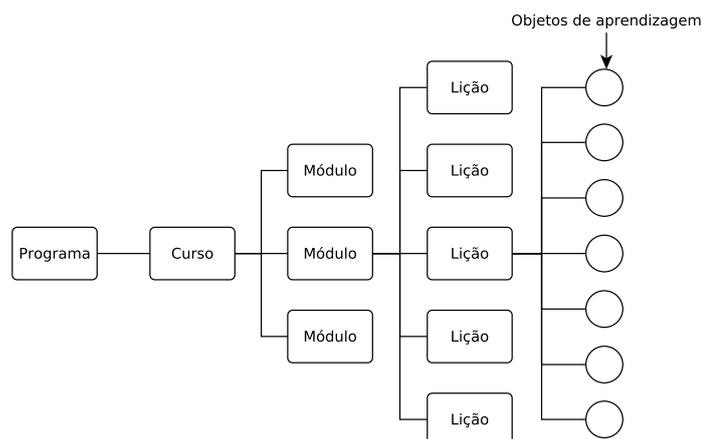


Figura 3.12 – Exemplo de taxonomia utilizada para objetos de aprendizado em um LMS.

A ideia central do uso de objetos de aprendizagem é caracterizada pela descoberta, reutilização e interoperabilidade. Para apoiar a capacidade de **descoberta**, os objetos de aprendizado são descritos por Metadados de Objetos de Aprendizado, formalizados como IEEE 1484.12 Metadados de Objetos de Aprendizado³. Para dar suporte à **reutilização**, o *IMS Consortium* propôs uma série de especificações, como o pacote *IMS Content*. E, para apoiar a **interoperabilidade**, a organização *Advanced Distributed Learning* das forças armadas dos EUA criou o Modelo de Referência de Objetos de Conteúdo Compartilhado. Os objetos de aprendizado foram projetados para reduzir o custo do aprendizado, padronizar o conteúdo do aprendizado e permitir o uso e reutilização do conteúdo do aprendizado pelos sistemas de gerenciamento de aprendizagem⁴.

3.10 Modelo de Referência de Objeto de Conteúdo Compartilhável

O Modelo de Referência de Objeto de Conteúdo Compartilhável, ou *Sharable Content Object Reference Model* (SCORM)⁵, é uma coleção de padrões e especificações para a tecnologia educacional eletrônica baseada na Web (também chamada de *e-learning*). Ele define as comunicações entre o conteúdo do cliente e um sistema *host*,

³ Para maiores detalhes, consultar <https://standards.ieee.org/standard/1484_12_1-2002.html>

⁴ De acordo com o *Learning Objects Metadata Workgroup*, objetos de aprendizagem podem ser definidos como qualquer entidade, digital ou não, que possa ser utilizada, reutilizada e/ou referenciada durante o aprendizado.

⁵ Para maiores detalhes, consultar <<https://eduworks.com/LOTT/Tutorial/index.html>>

geralmente suportado por um sistema de gerenciamento de aprendizado. O SCORM também define como o conteúdo pode ser empacotado em um arquivo transferível chamado *Package Interchange Format*.

O SCORM 2004 introduziu a ideia de sequenciamento, representada por um conjunto de regras que especifica a ordem na qual um aluno pode experimentar objetos de conteúdo. Em termos simples, eles restringem o aluno a um conjunto fixo de caminhos através do material de treinamento, permitem que o aluno marque seu progresso ao fazer pausas e asseguram a aceitabilidade das notas de teste obtidas. O padrão usa XML e é baseado nos resultados do trabalho realizado pela AICC, IMS Global, IEEE e Ariadne. O SCORM é composto por três subespecificações. O pacote de **conteúdo** especifica como o conteúdo deve ser empacotado e descrito. O **tempo** de execução especifica como o conteúdo deve ser iniciado, como os dados se comunicam com o sistema de *e-learning* e inclui as especificações para o modelo de dados desta comunicação. Por fim, o **sequenciamento** especifica como um aluno pode navegar entre partes de um curso (SCO). O SCORM é composto de 3 livros de referência (KRAUSS, 2016):

- Modelo de agregação de conteúdo (CAM) SCORM para gerenciar o conteúdo de aprendizado, lidar com sua estrutura, pesquisa e descoberta, bem como a definição de empacotamento e transferência de conteúdo
- Sequenciamento e navegação SCORM (SN) para lidar com sequências de conteúdo, atividades de aprendizado e dados de navegação
- SCORM *Run-Time Environment* (RTE) para a definição de requisitos para o LMS exibir Conteúdo SCORM

O maior benefício do SCORM é a **interoperabilidade**. No papel de criador de conteúdo o SCORM facilita a padronização para que o conteúdo seja carregado no sistema de *e-learning*. O SCORM garante que o conteúdo seja compatível com o sistema de *e-learning* para que o tal sistema possa importar, iniciar e rastrear os conteúdos criados com o padrão.

3.11 Repositórios de Objetos de Aprendizagem

Um Repositório de Objetos de Aprendizagem, ou *Learning Objects Repository* (LOR), é uma biblioteca *on-line* para armazenar, gerenciar e compartilhar seus recursos de aprendizagem (objetos de aprendizagem). Um objeto de aprendizado pode ser um questionário, uma apresentação, uma imagem, um vídeo ou qualquer tipo de artefato usado para os materiais de aprendizado. A Figura 3.13 mostra um exemplo de um repositório.

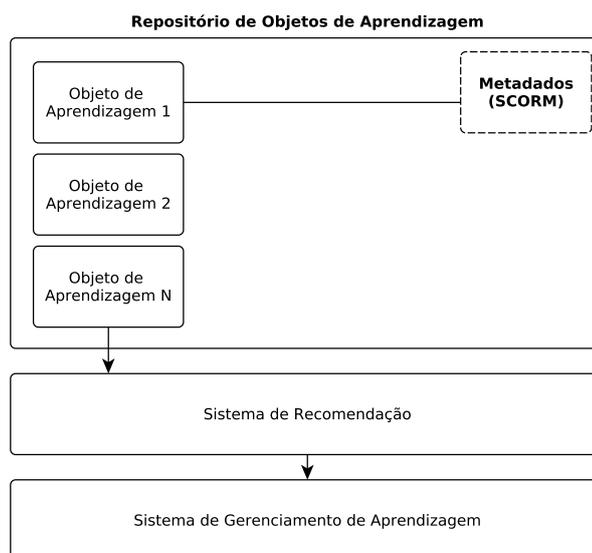


Figura 3.13 – Repositórios de Objetos de Aprendizagem são como depósitos virtuais onde ficam armazenados os materiais com fins educacionais para um sistema de *e-learning*.

Ao publicar os objetos de aprendizado no repositório, deve-se marcá-los com metadados para classificar e organizar seus objetos de aprendizado e torná-los facilmente pesquisáveis por outras pessoas ou sistemas. Podem-se classificar os objetos de aprendizagem de acordo com padrões ou objetivos de aprendizagem definidos. Você também pode revisar, classificar e fornecer *feedback* sobre os objetos de aprendizado para garantir recursos para os sistemas de *e-learning* de alta qualidade.

3.12 Estilos de Aprendizagem

De acordo com Felder e Silverman (FELDER; SILVERMAN et al., 1988), os estilos de aprendizagem (parte do estilo cognitivo) descrevem as preferências dos alunos sobre como uma matéria é apresentada, como trabalhar com esta matéria e como internalizar (adquirir, processar e armazenar) informações (FELDER; SILVERMAN et al., 1988). De acordo com Willingham (WILLINGHAM; HUGHES; DOBOLYI, 2015; WILLINGHAM, 2019), os alunos podem ter diversas maneiras preferidas de aprender. Assim, conhecer um estilo de aprendizagem do aluno pode ajudar a encontrar a maneira mais adequada de melhorar o processo de aprendizagem. Pode-se caracterizar o aprendizado como o procedimento pelo qual as informações são codificadas e armazenadas na memória de longo prazo. Por exemplo, personalizar o conteúdo de acordo com os estilos de aprendizagem dos alunos foi visto como útil para o aprendizado de diferentes maneiras, por exemplo, melhorando a realização, aumentando os resultados do aprendizado e diminuindo o tempo de aprendizado (BERNARD et al., 2017). Várias pesquisas em sistemas de aprendizagem sugerem que a aprendizagem dos alunos melhorasse quando o estilo de ensino dos instrutores

corresponde aos estilos de aprendizagem dos alunos (NORMADHI et al., 2019).

De acordo com Bernard et al. (BERNARD et al., 2017), existem vários métodos para classificar os estilos de aprendizagem. Um dos mais conhecidos, dentre eles, é o Modelo de Estilos de Aprendizagem Felder-Silverman (FSLSM)⁶. Este modelo propõe quatro dimensões (percepção, entrada, processamento e entendimento) para classificar os estilos de aprendizagem sendo que cada dimensão pode variar em um intervalo fechado de $[-11 : 11]$. A primeira dimensão deste modelo é **Processamento**, sendo as polaridades Ativo/Reflexivo, que determina se alguém prefere primeiro experimentar um assunto e depois argumenta sobre ele (Ativo); ou raciocine primeiro e depois experimente o assunto (Reflexivo). A segunda dimensão, **Percepção**, indicadas por Sensitivo/Intuitivo, determina se alguém prefere tocar em coisas para aprender (Sensitivo) ou observar coisas para induzir informações (Intuitivo). A terceira dimensão, **Entrada** relacionada a Visual/Verbal, determina se alguém prefere ver gráficos, tabelas, figuras (Visual) em vez de ler ou ouvir textos (Verbal), ou pelo contrário. Finalmente, a dimensão **Entendimento** Sequencial/Global determina se alguém prefere obter as informações de maneira sucessiva, aprendendo passo a passo (Sequencial) ou para obter um esboço das informações primeiro e depois ir aos detalhes, sem uma sequência predefinida (Global). A Tabela 3.3 exibe um resumo das dimensões, estilo e descrição para cada uma das dimensões do Modelo de Estilo de Aprendizagem Felder-Silverman (FSLSM).

Tabela 3.3 – Modelo de estilos de aprendizagem de Felder-Silverman

Dimensão	Estilo		Descrição
Percepção	Sensitivo	Intuitivo	Sensitivos preferem atividades concretas, Intuitivos preferem conceitos
Entrada	Visual	Verbal	Visual prefere imagens, Verbal preferem textos ou sons
Processamento	Ativo	Reflexivo	Ativos preferem tarefas práticas, Reflexivos pensar a respeito
Entendimento	Sequencial	Global	Sequencial preferem uma ordem estabelecida, Global a visão geral

De acordo com Bernard et al. (BERNARD et al., 2017), a caracterização dos estilos de aprendizagem é um problema que lida, simultaneamente, com muitos descritores e variáveis alvo. Os descritores podem surgir de várias fontes, como registros, questionários e bancos de dados (BERNARD et al., 2017). Além disso, os descritores geralmente estão associados a objetos de aprendizagem, como fóruns, conteúdos, esboços, questionários, auto-avaliações, exemplos e outros tipos de recursos. Os alvos são utilizados para permitir a compreensão do estilo de aprendizagem como resultado de uma combinação de descritores, que podem indicar se um aluno pode ser classificado como Ativo/Reflexivo, Sensitivo/Intuitivo, Visual/Verbal ou Sequencial/Global, com base em sua abordagem para reconhecer, processar e armazenar informações. Este problema é relevante porque é o primeiro passo para adequar a condição cognitiva para melhorar o aprendizado em relação aos sistemas de *e-learning* (NORMADHI et al., 2019)⁷.

⁶ Para maiores detalhes, consultar <<https://www.webtools.ncsu.edu/learningstyles/>>

⁷ Estas dimensões não estão dispostas em uma sequência; pelo contrário, estão todas conectadas com todas, como um grafo completo.

Parte II

Contribuições

4 Materiais e Métodos

Este capítulo apresenta os detalhes de dois experimentos realizados com arquiteturas de redes neurais artificiais para Aprendizado Adaptativo em Sistemas *E-learning*. O primeiro contém uma rede neural com arquitetura *feedforward* de múltiplas camadas, *Multi Layer Perceptron* (MLP), e de múltiplas saídas com aprendizado supervisionado para a classificação de estilos de aprendizagem. O segundo experimento apresenta variações da arquitetura *Deep Auto Encoder* (DAE) para fatoração de matrizes em um espaço latente com aprendizado semi-supervisionado para a recomendação de objetos de aprendizagem. Adicionalmente, questões comuns aos dois experimentos, como análise do conjunto de dados, variáveis descritivas e alvo, estratégia para otimização de hiperparâmetros, métricas de avaliação, técnicas para divisão do conjunto de dados e, também, métodos para treinamento e testes são discutidos com um nível de detalhes adequados para que os experimentos possam ser reproduzidos.

4.1 Classificação

Neste experimento um conjunto de dados rotulados foi utilizado para o treinamento de uma rede neural artificial baseada na arquitetura *Multi Layer Perceptron* (MLP). O aprendizado supervisionado foi empregado com a divisão do conjunto de dados feito via *Leave-One-Out Cross-Validation* (LOOCV), a função de perda (*Loss Function* (LF)) utilizada foi a *Root Mean Square Error* (RMSE) e as métricas de avaliação, baseadas na matriz de confusão, foram apresentadas e discutidas.

4.1.1 Conjunto de Dados

O processo de integração de estilos de aprendizagem no sistema de aprendizagem adaptável pode ser dividido em duas áreas principais; (i) previsão de estilos de aprendizagem e a (ii) aplicação deste modelo ao sistema de aprendizagem adaptável (BERNARD et al., 2017). O desenvolvimento começa com a escolha da estrutura dos estilos de aprendizagem. Isto é seguido pela determinação das fontes de dados, dos atributos dos estilos de aprendizagem e pela seleção do algoritmo de classificação. Após a avaliação, os modelos de classificação adequados e seus resultados são aplicados à diferentes aspectos do sistema adaptativo de *e-learning*, como a classificação de um estilo de aprendizagem ou recomendação de um conteúdo.

A primeira etapa para construir um modelo, baseado em algoritmos de inteligência computacional, reside em coletar e preparar o conjunto de dados. Os comportamentos

dos alunos foram coletados de um *Learning Management System* (LMS) desenvolvido especificamente para este experimento. Os 26 descritores foram baseados nos modelos de Sheeba e Krishnan (SHEEBA; KRISHNAN, 2019) e Bernad et al. (BERNARD et al., 2017). Estes descritores foram agrupados em 9 objetos de aprendizagem que são apresentados ao aluno em um curso dentro do LMS. O conjunto de dados é composto de três tipos de medidas; (i) contagem, que representa o número de vezes que um aluno visita um objeto de aprendizagem, (ii) tempo, que representa o tempo que o aluno passa em um objeto de aprendizagem, e (iii) binário, que representa os resultados do aluno ao responder às perguntas de um questionário. Estes registros foram coletados durante 15 dias e, para agrupar todos os resultados obtidos pelos alunos, cada descritor foi representado pela média dos registros de acesso dos alunos.

4.1.1.1 Variáveis Descritivas

A Tabela 4.1 mostra os descritores que foram coletados no LMS. Estes descritores também são considerados variáveis independentes para a construção do modelo.

Tabela 4.1 – Variáveis que capturam a interação aluno/objeto (BERNARD et al., 2017)

Objeto	Comportamento	Medida	Descrição
Content	content_visit	Quantidade	Número de vezes que o aluno visita um conteúdo
	content_stay	Tempo	Tempo gasto (em segundos) no conteúdo
Outline	outline_stay	Tempo	Tempo gasto (em segundos) no esboço
	outline_visit	Quantidade	Número de vezes que o aluno visita um esboço
Self-assessment	selfass_visit	Quantidade	Número de vezes que o aluno visita uma auto-avaliação
	selfass_stay	Tempo	Tempo gasto (em segundos) na auto-avaliação
	selfass_twice_wrong	Binário	Se o aluno testar a auto-avaliação
Exercise	exercise_visit	Quantidade	Número de vezes que o aluno visita um exercício
	exercise_stay	Tempo	Tempo gasto (em segundos) no exercício
Quiz	quiz_stay_results	Tempo	Tempo gasto (em segundos) no resultado do questionário
	quiz_revisions	Quantidade	Número de vezes que o aluno visita uma revisão de questionário
Forum	forum_visit	Quantidade	Número de vezes que o aluno visita um fórum
	forum_post	Quantidade	Número de vezes que o aluno postou em um fórum
	forum_stay	Tempo	Tempo gasto (em segundos) no fórum
Questions	ques_detail	Quantidade	Tempo gasto (em segundos) nos detalhes das perguntas
	ques_facts	Quantidade	Tempo gasto (em segundos) em perguntas dos fatos do tipo
	ques_concepts	Quantidade	Tempo gasto (em segundos) em perguntas do conceito de tipo
	ques_develop	Quantidade	Tempo gasto (em segundos) em perguntas dos fatos do tipo
	ques_graphics	Quantidade	Tempo gasto (em segundos) em perguntas dos gráficos do tipo
	ques_text	Quantidade	Tempo gasto (em segundos) em perguntas do tipo texto
	ques_overview	Quantidade	Tempo gasto (em segundos) na visão geral das perguntas
Navigation	ques_interpret	Quantidade	Tempo gasto (em segundos) na interpretação das perguntas
	navigation_skip	Quantidade	Tempo gasto (em segundos) em perguntas do tipo interpretar
	navigation_overview_visit	Quantidade	Número de vezes que o aluno pula a navegação
Example	navigation_overview_stay	Tempo	Número de vezes que o aluno visita a visão geral da navegação
	example_stay	Tempo	Tempo gasto (em segundos), por exemplo

A escolha das variáveis que descrevem a interação dos alunos com os objetos de aprendizagem é um ponto importante. Se somente objetos associados a uma única dimensão, por exemplo, processamento, forem apresentados aos alunos a captura do comportamento ficará prejudicada. Alguns autores sugerem o uso de algumas variáveis, que medem a quantidade de acesso e o tempo gasto, para identificar a interação entre os alunos e os objetos de aprendizagem (GRAF; KINSHUK; LIU, 2009; BERNARD et al.,

2017). Como as variáveis, descritas na Tabela 4.1, estão fora de escala, os dados devem ser normalizados para a entrada da rede neural (SRIVASTAVA et al., 2014). Existem alguns algoritmos para realizar esta tarefa. Segundo Khan (KHAN; HAYAT; PORIKLI, 2019), não existem diferenças substanciais entre eles, sendo mais uma questão de preferências. Neste caso, o algoritmo escolhido foi o *min/max*, representado pela Equação 4.1, pela simplicidade.

$$x'_i = \frac{x_i - \min(x_i)}{\max_i(x_i) - \min_i(x_i)} \quad (4.1)$$

O resultado da Equação 4.1 transforma todas as variáveis para a mesma escala sem a perda de informação. Isto é importante para o treinamento da rede, pois o valor de uma escala maior pode, de forma equivocada, gerar um peso maior para uma das características. A amplitude das variáveis, dos dados originais e normalizados, é mostrada na Figura 4.1.

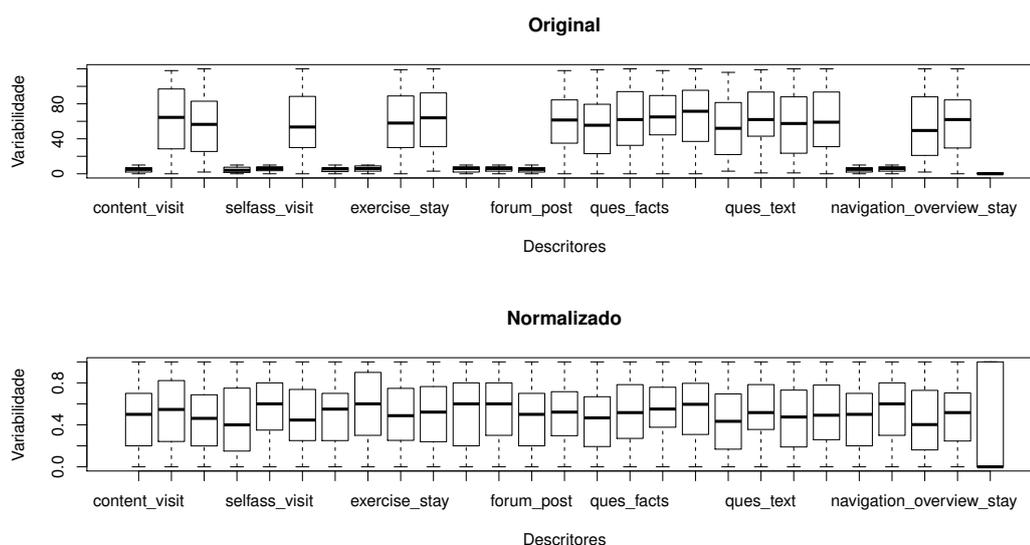


Figura 4.1 – A variabilidade dos comportamentos dos alunos para os objetos de aprendizagem. Na primeira figura, os dados não estão normalizados. Na segunda figura, os dados foram normalizados de acordo com a Equação 4.1 para aumentar a acurácia do modelo.

A variáveis do tipo *tempo* possuem um limitador de sessão de 120 segundos, para evitar eventos que não expressem a realidade devido ao fato de um aluno abandonar o LMS. Além disso, ao acessar o objeto de aprendizagem novamente, o tempo seria acumulado com o resultado expresso pela mediana.

4.1.1.2 Variáveis Alvo

O conjunto de dados resultante não fornece uma descrição de um estilo de aprendizagem para cada aluno. Esta informação é necessária para treinar um algoritmo baseado

no aprendizado supervisionado (BERNARD et al., 2017). Para superar esse problema, utilizou-se a adaptação do questionário Felder-Silverman para coletar o estilo de aprendizagem de cada aluno. Este questionário classifica um aluno no FSLSM usando as quatro dimensões; (i) **Processamento** (ativo/reflexivo), (ii) **Percepção** (sensitivo/intuitivo), (iii) **Entrada** (visual/verbal) e (iv) **Entendimento** (sequencial/global). Essa classificação é construída definindo um intervalo para cada dimensão (por exemplo, processamento) de $[-11 : 0]$ (ativo) a $(0 : 11]$ (reflexivo) e assim por diante. Os rótulos do conjunto de dados são mostrados na Tabela 4.2. Estes rótulos são considerados variáveis dependentes no modelo.

Tabela 4.2 – Variáveis Alvo

Dimensão	Variável	Medida	Descrição
Processamento	active_reflective	Quantidade	Tendência variando entre $[-11$ (active) e 11 (reflective)]
Percepção	sensing_intuitive	Quantidade	Tendência variando entre $[-11$ (sensing) e 11 (intuitive)]
Entrada	visual_verbal	Quantidade	Tendência variando entre $[-11$ (visual) e 11 (verbal)]
Entendimento	sequential_global	Quantidade	Tendência variando entre $[-11$ (sequential) e 11 (global)]

As variáveis alvo foram analisadas para identificar desbalanceamentos das classes presente nos dados. Ambas as polaridades, das quatro dimensões, foram analisadas. Primeiro identificou-se a variação entre as polaridades de cada dimensão e, depois, a distribuição de preferências ao longo de cada polaridade. Esta análise é mostrada na Figura 4.2.

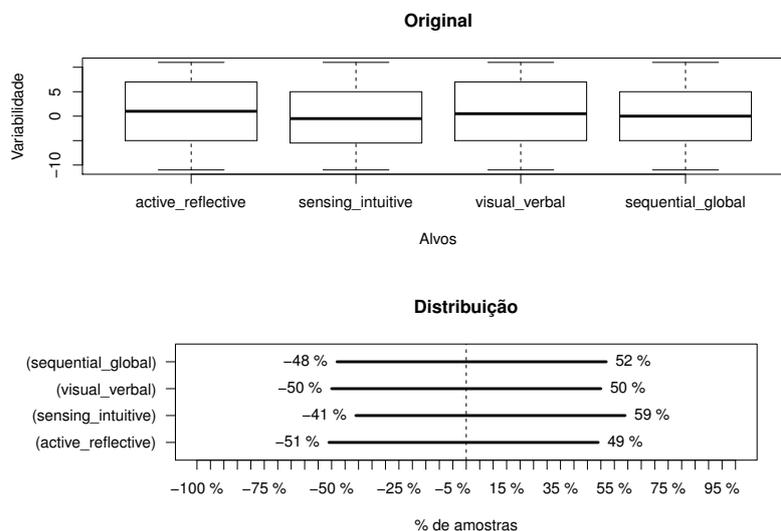


Figura 4.2 – A distribuição da preferência dos alunos entre $[-11 : 11]$ para cada dimensão. Não são determinísticas, e sim uma tendência (FELDER; SILVERMAN et al., 1988).

Pode-se notar na Figura 4.2 que os dados não apresentam desbalanceamento relacionado a cada uma das 4 dimensões. Por exemplo, na dimensão percepção, existe um volume maior de alunos com a preferência para intuitivo. Ver Seção 3.12 para maiores

detalhes e o Anexo D para uma visão de um único aluno. Apesar disso, essa diferença não é considerável e o fenômeno de dados desbalanceados não ocorreu no conjunto de dados de forma expressiva.

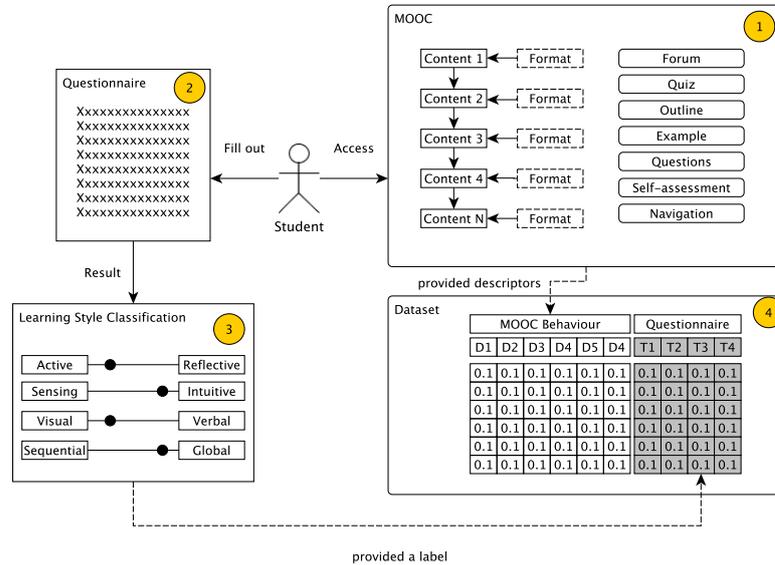


Figura 4.3 – O processo utilizado para a rotulação dos dados. Os dados foram obtidos pela adaptação do formulário de Felder-Silverman. (FELDER; SILVERMAN et al., 1988).

O processo para rotulação dos dados foi realizado como mostrado na Figura 4.3. O primeiro passo foi realizado coletando os dados da interação aluno/objeto no LMS. Esta interação gerou as observações dos descritores discutidos na Seção 4.1.1.1. Depois disso, no passo 2, os mesmos alunos preencheram o formulário de identificação de estilos de aprendizagem fornecidos pelos autores Felder-Silverman (FELDER; SILVERMAN et al., 1988). No passo 3, estes resultados foram tabulados e anexados ao conjunto de dados com os descritores; como exibido no passo 4. Um exemplo deste formulário pode ser visto no Anexo D. Além disso, um pré-processamento foi feito para relaxar o problema de classificação (HEIDARI et al., 2020). Como as classes indicam tendência, as mesmas foram transformadas para variáveis binárias, como mostram as Equações 4.2 à 4.5.

$$active_reflective = active_reflective < 0 \quad (4.2)$$

$$sensing_intuitive = sensing_intuitive < 0 \quad (4.3)$$

$$visual_verbal = visual_verbal < 0 \quad (4.4)$$

$$sequential_global = sequential_global < 0 \quad (4.5)$$

4.1.2 Multi Layer Perceptron

O algoritmo escolhido para a previsão de múltiplos alvos foi a Rede Neural Artificial de Múltiplas Camadas (RNA) por 5 razões; (i) há evidências de que este algoritmo é mais adequado para resolver problemas de classificação de estilos de aprendizagem (BERNARD et al., 2017), (ii) como muitos autores usam este mesmo algoritmo, pode-se comparar os resultados desta pesquisa com outros publicados (NORMADHI et al., 2019), (iii) a RNA funciona bem com conjuntos de dados bastante pequenos, o que é importante para esta linha de pesquisa, considerando que os conjuntos de dados típicos são bastante pequenos, (iv) o problema pode ser facilmente traduzido para a estrutura de rede de uma RNA, e (v) a RNA permite múltiplas saídas analisadas ao mesmo tempo. Além disso, a arquitetura da RNA empregada é a *perceptron* de múltiplas camadas, significando que tal rede possui uma ou mais camadas ocultas (KOLB; KOLB, 2005).

As camadas ocultas atuam como detectores de recursos e são necessárias em espaços de busca não separáveis linearmente; como tal, elas desempenham um papel crítico na operação de um *perceptron* multicamada. À medida que o processo de aprendizado progride no *perceptron* multicamada, os neurônios ocultos começam a descobrir gradualmente os recursos que caracterizam os dados de treinamento. Eles fazem isto executando um processamento não linear nos dados de entrada e transformando-os em um novo espaço chamado espaço de recurso. Neste novo espaço, as classes de interesse em uma tarefa de classificação de padrões, por exemplo, podem ser mais facilmente separadas umas das outras do que no espaço de dados de entrada original. De fato, é a formação desse espaço de recurso através do aprendizado supervisionado que distingue o *perceptron* multicamada do *perceptron*. A literatura sugere que o número de camadas ocultas deve estar entre o $\log T$ (onde T é o tamanho do conjunto de treinamento) e $2 \times$ o número de entradas (BERNARD et al., 2017).

Um método popular para o treinamento de *perceptron* multicamada é o algoritmo de retropropagação, que inclui o algoritmo LMS como um caso especial. O treinamento prossegue em duas fases. Na primeira fase, denominada fase direta, os pesos sinápticos da rede são fixos e o sinal de entrada é propagado através da rede, camada por camada, até atingir a saída. Assim, nesta fase, as alterações são confinadas aos potenciais e saídas de ativação dos neurônios na rede. Na segunda fase, chamada fase reversa, é produzido um sinal de erro comparando a saída da rede com a resposta desejada. O sinal de erro resultante é propagado pela rede, novamente camada por camada, porém desta vez a propagação é realizada na direção inversa. Nesta segunda fase, ajustes sucessivos são feitos nos pesos sinápticos da rede. O cálculo dos ajustes para a camada de saída é direto, mas é muito mais desafiador para as camadas ocultas.

4.1.3 Hiperparâmetros

Nesta seção os hiperparâmetros utilizados para a construção do modelo são apresentados e discutidos. Além disso, as estratégias para otimização dos mesmos são abordadas.

4.1.3.1 Taxa de Aprendizado

A taxa de aprendizado η , ou *Learning Rate*, é um número real e **constante** que determina o quanto os pesos da rede neural podem ser alterados quando for observado um erro entre o valor esperado e o valor retornado durante o treinamento. Este valor deve ser selecionado cuidadosamente, uma vez que, quando for muito baixo, a rede tomará mais tempo para convergir e, quando alto, a rede pode nunca convergir. Entre o tempo de convergência e a possibilidade de não convergência, optou-se por correr o risco de maior tempo de convergência e manteve-se a taxa de aprendizado em 0,001 (HEIDARI et al., 2020).

4.1.3.2 Momentum

Embora o algoritmo de *backpropagation* seja comumente usado para o processo de aprendizagem de redes neurais, ele sofre de baixa velocidade para treinamento e, dentre as maneiras de acelerá-lo, é possível incluir um coeficiente denominado **momentum** em uma de suas equações, conforme mostrado por (ALRAIMI, 2015). A utilização do *momentum* para o problema em questão não trouxe melhoras na acurácia e no tempo de processamento. Isto deve-se ao fato do conjunto utilizado para treinamento e testes ser pequeno (100 observações). A Equação 4.6 mostra como o treinamento é usualmente feito aplicando a taxa de aprendizagem η .

$$\Delta w_{ij} = \left(\eta \times \frac{\partial E}{\partial w_{ij}} \right) \quad (4.6)$$

Por sua vez, ao optar por usar o hiperparâmetro *momentum* γ , deve-se modificar a Equação 4.6 para acomodá-lo. O resultado pode ser visto na Equação 4.7.

$$\Delta w_{ij} = \left(\eta \times \frac{\partial E}{\partial w_{ij}} \right) + (\gamma \times \Delta w_{ij}^{t-1}) \quad (4.7)$$

Optou-se em não utilizar o hiperparâmetro *momentum* visto que não trouxe benefícios para o problema, como citado anteriormente. Além disso, a adição de um hiperparâmetro aumentaria a complexidade da solução.

4.1.3.3 Quantidade de Épocas

Uma época é um ciclo percorrido por toda a base de treinamento (HEIDARI et al., 2020). Para a configuração padrão da MLP, verificou-se o comportamento da eficiência ao ajustar a quantidade de épocas para realização do processo de treinamento. Foram testados vários valores de épocas entre 1 e 200. No geral, o valor de 50 iterações foi suficiente para o treinamento, como pode ser observado na Figura 4.4.

4.1.3.4 Quantidade de Neurônios Ocultos

Neurônios ocultos são aqueles entre os neurônios de entrada e os de saída. A quantidade de neurônios ocultos não é fácil de se determinar, sendo necessário experimentação, uma vez que este hiperparâmetro é dependente dos tipos de dados e/ou do problema em si. Como estratégia adotou-se a variação da quantidade de camadas comparadas com a acurácia do modelo. A quantidade de camadas ótimas para o problema foram duas camadas escondidas, a primeira com x e a segunda com y neurônios, respectivamente. As camadas de entrada representam as variáveis da Tabela 4.1, portanto 26 neurônios. A camada de saída, por sua vez, representam os neurônios da Tabela 4.2, portanto 4 neurônios. O modelo resultante pode ser visto na Figura 5.6.

4.1.4 Divisão do Conjunto de Dados

O método *Leave-One-Out Cross Validation* (LOOCV) é um caso especial do k -fold para $K = n$, ou seja, a cada iteração, a amostra de validação será correspondente a uma observação $d_k = (y_k, x_k)$, $k = 1, 2, \dots, n$, e a amostra de treino para criar o preditor é feita com as outras $n - 1$ observações, onde usaremos a notação $d(-k)$, ou seja, é o conjunto de todas as observações exceto a k -ésima. A estimativa do risco esperado pode ser definida como (Equação 4.8):

$$LOOCV = \frac{1}{n} \sum_{k=1}^n (y_k, \hat{f}_{(-k)}(\mathbf{x}_k)) \quad (4.8)$$

onde $\hat{f}_{(-k)}(\mathbf{x}_k)$ é o preditor criado em cada iteração k do processo, retirando-se a observação (y_k, x_k) da amostra de treino. Portanto, o LOOCV é uma técnica para divisão do conjunto de dados com baixo viés, pois a amostra de treino é quase a base toda, principalmente quando n é grande. Porém, LOOCV tem alta variabilidade, pois as parcelas de cada etapa possuem apenas uma observação para teste. Além disso, por treinar n vezes o modelo, o custo computacional deste método pode ser elevado se for necessário uma amostra muito grande. Neste caso optou-se pelo LOOCV, pois o número de observações é pequeno e permite com que o viés do modelo seja reduzido. Maiores detalhes sobre o LOOCV podem ser obtidos na Seção 3.8.2.

4.1.5 Métricas de Avaliação

Uma matriz de confusão, ou *confusion matrix* (CM), contém informações sobre classificações reais e previstas feitas por um sistema de classificação. O desempenho de tais sistemas é geralmente avaliado usando os dados na matriz. A Tabela 4.3 a seguir mostra a matriz de confusão para um classificador de duas classes (classificador binário).

Tabela 4.3 – Matriz de Confusão para um classificador binário

		Esperado		Total
		Positivo	Negativo	
Predito	Positivo	TP	FP	$TP + FP$
	Negativo	FN	TN	$FN + TN$
Total		$TP + FN$	$FP + TN$	N

As entradas na matriz de confusão têm o seguinte significado no contexto dessa pesquisa. Vários termos padronizados foram definidos para a matriz de 2 classes (binária):

- TP é o número de previsões corretas de que uma instância é negativa,
- FP é o número de previsões incorretas de que uma instância é positiva,
- FN é o número de previsões incorretas que uma instância é negativa e
- TN é o número de previsões corretas de que uma instância é positiva.

A taxa negativa verdadeira (TN) é definida como a proporção de casos negativos que foram classificados corretamente, calculados usando a Equação 4.9.

$$Sensitivity = \frac{TP}{TP + FP} \quad (4.9)$$

A taxa de falsos positivos (FP) é a proporção de casos negativos que foram incorretamente classificados como positivos, calculados usando a Equação 4.10.

$$Specificity = \frac{TN}{FN + TN} \quad (4.10)$$

A prevalência esta relacionada com a frequência que determinada classe acontece no conjunto de dados, sendo representado pela Equação 4.11.

$$Prevalence = \frac{TP + FP}{TP + FN + FP + TN} \quad (4.11)$$

Os valores preditivos positivos (PPV) são as proporções de resultados positivos em testes de diagnóstico que são verdadeiros positivos. Um resultado alto pode ser

interpretado como indicando a precisão dessa métrica; ele depende também da prevalência e é representado pela Equação 4.12.

$$PPV = \frac{sensitivity \times prevalence}{(sensitivity \times prevalence) + ((1 - specificity) \times (1 - prevalence))} \quad (4.12)$$

O NPV funciona de maneira semelhante ao PPV, entretanto, aplicado aos valores negativos e sendo representado pela Equação 4.13.

$$NPV = \frac{sensitivity \times (1 - prevalence)}{(1 - sensitivity) \times (prevalence) + ((1 - specificity) \times (1 - prevalence))} \quad (4.13)$$

A métrica *Detection Rate* apresenta a quantidade de verdadeiros positivo classificados pelo modelo, sendo representada pela Equação 4.14.

$$Detection\ Rate = \frac{TP}{TP + FN + FP + TN} \quad (4.14)$$

De maneira adicional à métrica *Detection Rate*, a *Detection Prevalence* apresenta a prevalência de classificações verdadeiro positivo realizadas, sendo representada pela Equação 4.15.

$$Detection\ Prevalence = \frac{TP + FN}{TP + FN + FP + TN} \quad (4.15)$$

A *Balanced Accuracy* é uma métrica que se pode usar ao avaliar a qualidade de um classificador binário. É especialmente útil quando as classes estão desequilibradas, ou seja, uma das duas classes aparece muito mais frequentemente que a outra. Isso acontece frequentemente em muitas situações, como a detecção de anomalias e a presença de uma doença. Esta métrica é representada pela Equação 4.16.

$$Balanced\ Accuracy = \frac{sensitivity + specificity}{2} \quad (4.16)$$

Adicionalmente, precisão (P) é a proporção dos casos positivos previstos que estavam corretos, conforme calculado usando a Equação 4.17.

$$Precision = \frac{TP}{TP + FN} \quad (4.17)$$

A *Recall* ou taxa positiva verdadeira (TP) é a proporção de casos positivos que foram identificados corretamente, conforme calculado usando a Equação 4.18.

$$Recall = \frac{TP}{TP + FP} \quad (4.18)$$

A *F1-score* métrica combina precisão e *Recall* de modo a trazer um número único que indique a qualidade geral do seu modelo e trabalha bem até com conjuntos de dados que possuem classes desproporcionais. A fórmula que define a métrica é representada pela Equação 4.19.

$$F1 = \frac{(1 + \alpha^2) \times precision \times recall}{(\alpha^2 \times precision) + recall} \quad (4.19)$$

Por fim, a acurácia (CA) é a proporção do número total de previsões corretas. É determinada usando a Equação 4.20 sendo que, quanto maior (ou próxima de 1), melhor o classificador.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.20)$$

Deve-se notar que estas métricas de avaliação devem ser interpretadas em conjunto e não individualmente. Por exemplo, uma acurácia com valor 1 e com as demais métricas com valores ineficientes, não significa que o classificador tem a capacidade de generalizar e prever, com uma eficácia de 100%, novas observações. Em outras palavras, não significa que o classificador terá uma **eficácia máxima**, em termos de verdadeiros positivos (TP) e verdadeiros negativos (TN), com relação a novas observações que não foram utilizadas para o treinamento e testes. Portanto, as métricas de avaliação devem ser combinadas e analisadas em conjunto para que sirvam de suporte à decisão de qual modelo, gerado pelo treinamento da rede neural artificial, utilizar.

4.1.6 Treinamento da Rede

O comportamento da função de perda (*loss function*) *Root Mean Square Error* (RMSE) do treinamento da rede MLP inicia com um valor alto e reduz, de forma logarítmica, conforme o avanço das épocas. A primeiras épocas tem uma redução substancial e conforme o avanço a redução vai diminuindo até a estabilização da rede. Este comportamento pode ser observado na Figura 4.4.



Figura 4.4 – Comportamento da função de perda RMSE do treinamento da rede MLP.

Caso os hiperparâmetros de inicialização estejam em uma faixa adequada, a tendência do comportamento é o representado pela Figura 4.4. Em outros caso a rede pode não convergir. Maiores detalhes desta métrica podem ser vistos na Seção 4.2.7.

4.2 Recomendação

Este experimento apresenta os detalhes para a reprodução da arquitetura de *Deep Auto Encoder* para criar uma assinatura de preferências de um determinado aluno para o conjunto de objetos de aprendizagem.

4.2.1 Conjunto de Dados

Este trabalho utiliza os dados das interações dos alunos de uma plataforma de cursos *on-line* com cada um dos cursos. Este conjunto de dados possui os dados do aluno, do objeto de aprendizagem (no caso o curso) e o tempo gasto neste curso. Um exemplo deste conjunto de dados pode ser visto na Tabela 4.4 a seguir.

Tabela 4.4 – Exemplo dos dados brutos obtidos do ambiente de cursos *on-line*

student_id	object_id	learning_object	hours	timestamp
1	1	Learning Object 1	1,10	1264275901
2	2	Learning Object 1	0,30	1264275901
...
n	n	Learning Object n	1,00	1264275901

Este conjunto de dados foi transposto para criar uma matriz de interação aluno/objeto, conforme exibido na Tabela 4.5. Deve-se notar que os pares aluno/objeto que não tiveram interação apresentam o valor nulo/ausente/desconhecido.

Tabela 4.5 – Transformada do conjunto de dados na matriz de interação aluno/objeto

	object_id				
		1	2	...	n
student_id	1	1,10		...	
	2		0,30	...	

	n				1,00

Um fenômeno que acontece em sistemas de recomendação educacionais está relacionado com o cenário onde alunos não acessam certos objetos de aprendizagem. Isto faz com que a interação aluno/objeto ($A \times O$) não seja realizada por *feedback* implícito ou explícito. A teoria da cauda longa é pautada em duas hipóteses fundamentalmente diferentes para explicar esse fenômeno. A primeira hipótese, e em geral a mais aceita (WANG; WANG; YEUNG, 2015), afirma que a maioria dos alunos segue o **comportamento de massa**,

enquanto que apenas uma minoria apresenta comportamentos especializados. A segunda hipótese afirma que a maioria dos alunos possui **caráter excêntrico** no sentido de consumir tanto objetos específicos quanto de massa (WANG; WANG; YEUNG, 2015).

Adicionalmente, Gobel e Mehm (GÖBEL; MEHM, 2013) apresentam evidências empíricas de que ambas as hipóteses são verdadeiras dependendo do tipo de objeto de aprendizagem apresentado. Enquanto no mercado de filmes, por exemplo, a primeira hipótese predomina, no mercado de músicas a segunda hipótese mostrou que as maiores avaliações foram dadas ao mercado de específico, demonstrando o potencial lucrativo deste mercado. Para fazer com que os alunos encontrem os objetos mais apropriado às suas preferências, os LMS utilizam os sistemas de recomendação para filtrar o vasto catálogo de objetos de aprendizagem disponíveis a fim de sugerir ao aluno somente os objetos mais relevantes as suas preferências dado seu histórico de utilização. Portanto, deve-se notar que o fenômeno de cauda longa também ocorre no sistema de *e-learning*, como mostra a Figura 4.5.

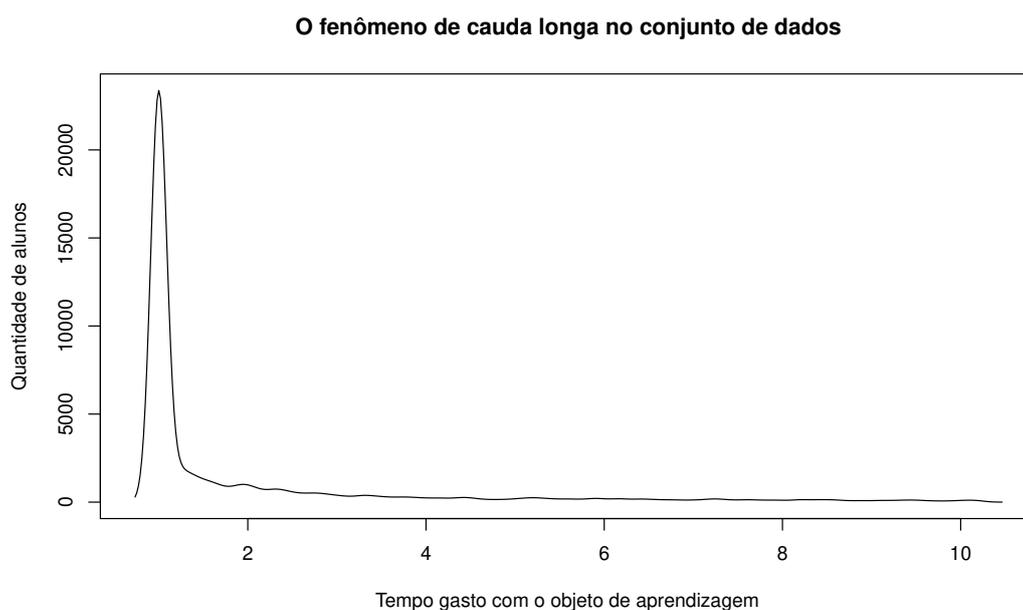


Figura 4.5 – O fenômeno de cauda longa para a quantidade de tempo gasto com os objetos de aprendizagem. A grande parte dos alunos gasta entre 0 e 2 horas em cada vídeo. Esta quantidade de alunos tem um decaimento logarítmico conforme o tempo aumenta o que representa que muitos alunos gastam pouco tempo nos objetos de aprendizagem.

Igualmente, o fenômeno ocorre para os alunos e objetos de aprendizagem. Existe uma quantidade de alunos que acessam frequentemente o sistema de *e-learning* enquanto a grande maioria tem baixa frequência de acesso. Por sua vez, os objetos de aprendizagem apresentam o mesmo comportamento, sendo que alguns são frequentemente acessados enquanto outros, raramente. Tal fenômeno é evidenciado pela Figura 4.8.

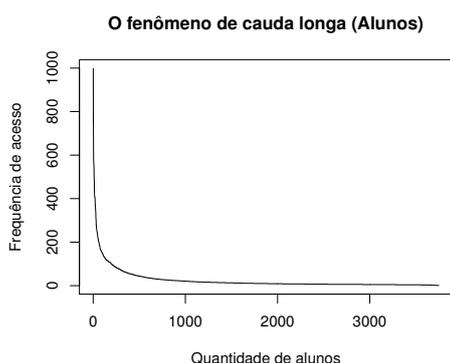


Figura 4.6 – Alunos

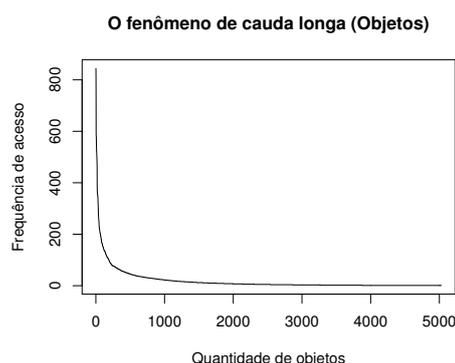


Figura 4.7 – Objetos

Figura 4.8 – O fenômeno de cauda longa também acontece para os demais elementos do conjunto de dados. Por exemplo, pode-se notar que a frequência de alunos (a) tem um comportamento logarítmico, sendo que poucos alunos acessam frequentemente os objetos de aprendizagem. Da mesma forma ocorre com os objetos de aprendizagem (b); poucos são acessados frequentemente.

Antes de discutir as principais aplicações dos sistemas de recomendação, deve-se ponderar o fenômeno da cauda longa ¹ que torna os sistemas de recomendação necessários. Os sistemas educacionais tradicionais são caracterizados pela **escassez de recursos**. Por outro lado, sistemas de *e-learning* podem disponibilizar para o aluno **tudo o que existe**. Assim, uma biblioteca física pode ter vários milhares de livros em suas prateleiras, mas uma biblioteca virtual oferece milhões de livros. Um jornal físico pode imprimir várias dezenas de artigos por dia, enquanto os serviços de indexação oferecem milhares por dia.

A recomendação no mundo físico é bastante simples. Primeiro não é possível personalizar a educação para cada aluno. Assim, a escolha do que é disponibilizado é governada apenas por restrições de recursos. Normalmente, uma biblioteca exibirá apenas os livros mais populares e um jornal imprimirá apenas os artigos que acredita que mais pessoas terão interesse. No primeiro caso, os números de vendas regem as escolhas, no segundo caso, o julgamento editorial serve. A distinção entre os mundos físico e virtual tem sido chamada o fenômeno da cauda longa, e é sugerido na Figura 4.8. O eixo vertical representa popularidade (o número de vezes que um item é escolhido). Os itens são ordenados no eixo horizontal de acordo com sua popularidade. Instituições físicas fornecem apenas os itens mais populares à esquerda da linha vertical, enquanto as instituições virtuais fornecem toda a gama de itens: a cauda bem como os itens populares. O fenômeno da cauda longa obriga as instituições virtuais a recomendar itens para usuários individuais. Não é possível apresentar todos os itens disponíveis ao usuário, da maneira que as instituições físicas podem. Portanto, o fenômeno de cauda longa é um dos fatores para um sistema

¹ Cauda longa pode ser apresentada com o nome de cauda pesada *heavy-tailed*. Na teoria da probabilidade, distribuições de cauda pesada (*heavy-tailed*) são distribuições de probabilidade cujas caudas não são exponencialmente delimitadas: ou seja, elas têm caudas mais pesadas que a distribuição exponencial.

de recomendação em *e-learning*. Pode-se adaptar o conteúdo, obrigatório e opcional, às preferências dos alunos sendo, em um caso extremo, **um sistema de *e-learning* para cada aluno**.

4.2.1.1 Objetos de Aprendizagem

Os objetos de aprendizagem utilizados foram vídeos sobre os mais variados conteúdos em um LMS para cursos livres. Este ambiente funciona como um concentrador de cursos de vários temas e tem a característica de permitir ao aluno, ao assistir um curso, também assistir outro relacionado. Desta forma, não existem restrições, como cursos obrigatórios; também não existem restrições de tempo, sequência, notas mínimas e frequência. Adicionalmente, estes objetos não possuem hierarquias ou agrupamentos de qualquer ordem, apenas similaridade entre conteúdos. O *feedback* utilizado para representar a interação aluno/objeto é a quantidade de tempo, medido em horas, utilizados em cada objeto. O aluno pode assistir ao vídeo completamente, parcialmente, em partes, ou de forma completa, porém fracionada.

4.2.1.2 Feedback

Os tipos de *feedback* podem ser representados por dois conjuntos: **implícitos** e **explícitos** (CHAE; KIM; LEE, 2019; CHEN; PENG, 2018; ZHAO et al., 2018). Os *feedback* implícitos podem variar de acordo com os dados disponíveis, sendo eles: a quantidade de visualização do objeto de aprendizagem, o tempo gasto com esse objeto, se o objeto foi marcado para visualização futura e se o objeto foi adicionado a uma lista de favoritos. Além disso, no contexto educacional, pode-se considerar, em caso de um objeto avaliativo, armazenar o resultado do aluno. Os *feedback* explícitos podem ser um sinal positivo ou negativo no objeto, pode ser um comentário sobre esse objeto, uma recomendação para um outro aluno. Atualmente, existem 10 categorias de *feedback* que podem ser utilizados em sistemas de *e-learning* (CHEN et al., 2020).

1. **Inicializado**, acionado quando um objeto de aprendizagem foi aberto ou um vídeo foi iniciado
2. **Encerrado**, é acionado quando um objeto de aprendizagem foi encerrado
3. **Abandonado**, acionado quando um aluno sai de um objeto de aprendizagem inesperadamente (por exemplo, fechando o navegador)
4. **Respondido**, acionando quando um aluno responde a uma pergunta ou fornece uma auto-avaliação
5. **Baixado**, ocorre quando um aluno realiza uma cópia do objeto de aprendizagem do computador remoto para o local; normalmente um PDF

6. **Suspenso**, é acionado quando um aluno pausa um vídeo antes do final
7. **Reiniciado**, é acionado quando um aluno retoma um vídeo pausado anteriormente
8. **Concluído**, é acionado quando um aluno assiste a um vídeo até o fim
9. **Dispensado**, é acionado quando um aluno avança o conteúdo em um vídeo (alterado a posição da barra de progresso manualmente)
10. **Comentado**, é acionado quando um aluno escreve um comentário na discussão

Deve-se notar que estes *feedback* não são únicos. Pode-se capturar alguns *feedback* originados de **requisições HTTP**, como por exemplo horário e local ². Portanto, o aprendizado adaptativo em sistemas de *e-learning*, suportados por TIC, podem utilizar tais recursos.

Para o conjunto de dados utilizado nesta pesquisa, o único *feedback* disponível é o tempo gasto, expresso em horas, em um objeto de aprendizagem. Entretanto, a utilização deste *feedback* sem nenhuma análise pode levar a equívocos gerados por eventos descuidados, como os citados na Seção 4.2.1.2. Desta forma, uma adaptação foi feita visando eliminar eventuais ruídos. Segundo o trabalho de (XUE et al., 2017; CHAE; KIM; LEE, 2019; CHEN; PENG, 2018), pode-se utilizar a seguinte escala com relação ao tempo de utilização de objetos de aprendizagem, conforme apresentado na Tabela 4.6.

Tabela 4.6 – Interpretação dos *feedback*

Duração	Interpretação
Menos que 5 segundos	Sem interesse
Mais que 5 segundos	Interessados
Mais que 1 minuto	Muito interessado
Mais que 5 minutos	Provavelmente parou para ver outra coisa
Mais que 10 minutos	Provavelmente abandonou o objeto de aprendizagem

Esta escala está relacionada com o tempo sem interação e/ou o tempo em que o vídeo não estava sendo executado. Portanto, pode-se associar os tempos dos eventos da Tabela 4.6 com as categorias apresentadas na Seção 4.2.1.2: menos que 5 segundos (Dispensado), mais que 5 segundos (Inicializado), mais que 1 minuto (Reiniciado, Encerrado, Concluído), mais que 5 minutos (Suspenso) e mais que 10 minutos (Abandonado). As categorias Respondido, Baixado e Comentado não se aplicam aos dados desta pesquisa.

4.2.1.3 Análise dos Dados

O conjunto de dados utilizado está resumido na Tabela 4.7. Estes dados representam a interação de 3.757 alunos com 5.104 objetos de aprendizagem disponíveis.

² A especificação completa pode ser vista no endereço <https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>

O *feedback* implícito está relacionado com o tempo gasto no objeto de aprendizagem. Isto representa uma matriz com a possibilidade de 19.175.728 interações únicas e uma esparsidade de aproximadamente 99% (maiores detalhes na Seção 4.2.2).

Tabela 4.7 – Resumo do conjunto de dados

Dados	Alunos	Objetos	Avaliação	Matriz	Esparsidade
Cursos	3757	5104	(0:10]	19.175.728	0.9939978

4.2.1.4 Pré-processamento

A característica considerada como *feedback* é *hours* (Tabela 4.4). Esta característica possui **ruídos** (*outliers*) que foram tratados pois não expressam a realidade; por exemplo um tempo acima de 10 horas gasto com um objeto de aprendizagem. Além disso, esta característica deve ser **normalizada** para ser processada como entrada da rede neural.

4.2.2 Matriz Esparsa

Uma matriz pode ser dita esparsa se seus elementos não-nulos estão concentrados ou distribuídos de forma não-sistemática. Isto é, uma matriz esparsa caracteriza-se pelo volume de **falta de informação** existente em uma matriz, muitas vezes representada por valores não definidos ou nulos (SILVA; JUNIOR; CALOBA, 2018). A esparsidade no sentido amplo (OS, *Overall Sparsity*) mede o nível de esparsidade de uma matriz $R_{n \times m}$ como um todo, conforme a Equação 4.21. Maiores detalhes sobre matrizes esparsas podem ser obtidos na Seção 3.6.

$$sparsity = 1 - \frac{total_elementos_maior_zero}{total_elementos} \quad (4.21)$$

Desta forma, quanto mais preenchida a matriz for, mais próximo de 0 tende a ser o resultado. Quanto menos preenchida, mais próximo de 1. Utilizando a Equação 4.21, o resultado da esparsidade calculado foi de 0.9939978, ou 99%. Isto representa um alto índice de esparsidade. Em sistemas de recomendação, a esparsidade também pode ser medida a nível de itens ou de usuários. Ao comparar-se um usuário-alvo com todos os outros, por exemplo, pode-se obter uma estimativa do quanto aquele usuário costuma avaliar seus itens consumidos quando comparados com outros. Assim, duas medidas de esparsidade no sentido específico são apresentadas (SILVA; JUNIOR; CALOBA, 2018). A esparsidade específica do usuário (USS, *User Specific Sparsity*) é definida conforme a Equação 4.22.

$$USS = 1 - \frac{n_u}{max_{u \in U}(n_u)} \quad (4.22)$$

Como já mencionado, os dados da interação aluno/objeto podem ser coletados explicitamente ou implicitamente. No primeiro caso, é necessária a participação do aluno que é solicitado a explicitamente enviar sua classificação para o item fornecido. Em contraste com isto, preferências implícitas são inferidas a partir das ações do aluno no contexto de um objeto (é por isso que o termo “interação aluno/objeto” é usado em vez da palavra “classificação” quando refere-se às preferências dos alunos). Os dados podem ser coletados implicitamente no lado do cliente ou do lado do servidor. No primeiro caso, o aluno utiliza um software do lado do cliente que registra suas ações. Esta possibilidade geralmente é omitida pois a distribuição do software tende a ser mais complexa. No segundo caso, o *log* é feito por um servidor. No contexto da *Web*, preferências implícitas podem ser determinadas a partir de *logs* de acesso que são automaticamente mantidos por servidores da *Web*. A esparsidade do item (no caso objeto) é calculada pela Equação 4.23.

$$ISS = 1 - \frac{n_i}{\max_{i \in I}(n_i)} \quad (4.23)$$

Pode-se notar a esparsidade da matriz ao visualizar a imagem das interações aluno/objeto. A Figura 4.9 apresenta tal interação. Os pontos mais escuros são os locais onde existem mais interações entre alunos e objetos de aprendizagem. Em outras palavras, são os alunos que mais acessam os objetos mais requisitados. Com isso, grande parte da matriz, representada por pontos mais claros, não são acessados, ou acessados com uma frequência muito pequena. Isto remete ao fenômeno de cauda longa discutida na Seção 4.2.1. Portanto, um sistema de *e-learning* adaptativo deve ser capaz de encontrar nas interações inexistentes, objetos de aprendizagem que possam ser do interesse de um determinado aluno (WANG; WANG; YEUNG, 2015).

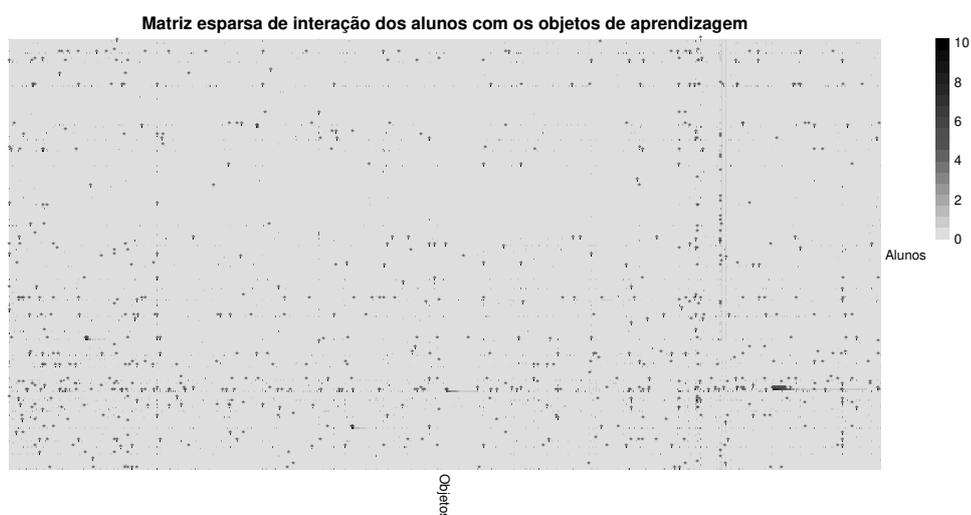


Figura 4.9 – A matriz esparsa resultante da transposição do conjunto de dados. Esta matriz indica que poucos alunos acessam poucos objetos de aprendizagem. Desta forma, têm-se um grande numero de objetos e de alunos que não foram beneficiados dos conteúdos que estão escondidos.

Os dados coletados são pré-processados e organizados em uma matriz de aluno/objeto. Linhas representam alunos e colunas representam objetos. Cada elemento da matriz está associado, em geral, a um conjunto de ações que um aluno executou no contexto de um objeto. Na maioria dos casos, um elemento da matriz é um número único que representa uma classificação **deduzida das ações do aluno**. Como este, geralmente, não acessa todos os objetos de aprendizagem do repositório de objetos de aprendizagem (ver Seção 3.11), o vetor (ou seja, a linha da matriz) representando o aluno está incompleto, faltando alguns ou muitos valores. Como visto na Figura 4.9, estas interações de aluno/objeto possuem características intrínsecas aos sistemas de recomendação. Uma delas é o já citado fenômeno de cauda longa. Isto ocasiona uma matriz com poucas interações entre os alunos e os objetos. Tal fenômeno aparece novamente na interação aluno/objeto, representado pelo tempo de acesso de um aluno a um objeto de aprendizagem. Isto corrobora o comportamento de poucos alunos acessam poucos objetos por pouco tempo. Isto é apresentado na Figura 4.12 (a) e (b) e pode ser comparado com a matriz apresentada na Figura 4.9.

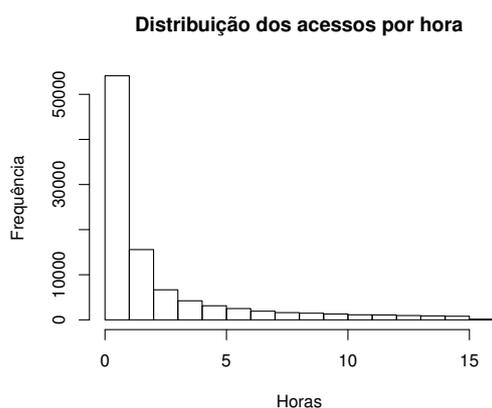


Figura 4.10 – Alunos

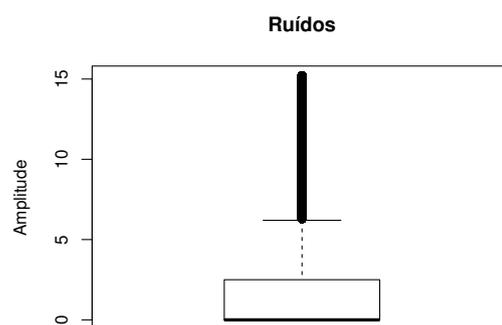


Figura 4.11 – Objetos

Figura 4.12 – (a) O fenômeno de cauda longa representado pela distribuição da frequência de acesso dos alunos pela quantidade de horas. (b) Os alunos que acessam objetos por mais de 5 horas são considerados como *outliers*.

A esparsidade da matriz pode ser visualizada, de forma complementar, utilizando a técnica chamada de *Principal Component Analysis* (PCA). O PCA é uma técnica para reduzir a dimensão de um conjunto de dados preservando, de certa forma, suas propriedades. Consiste em um procedimento matemático que utiliza uma transformação ortogonal para converter um conjunto de observações de variáveis possivelmente correlacionadas num conjunto de valores de variáveis linearmente não correlacionadas chamadas de componentes principais. O número de componentes principais é sempre menor ou igual ao número de variáveis originais. A Figura 4.13 apresenta o resultado evidenciando que as interações concentraram-se em um pequeno grupo de aluno/objeto.

Agrupamento da Matriz Aluno x Objeto

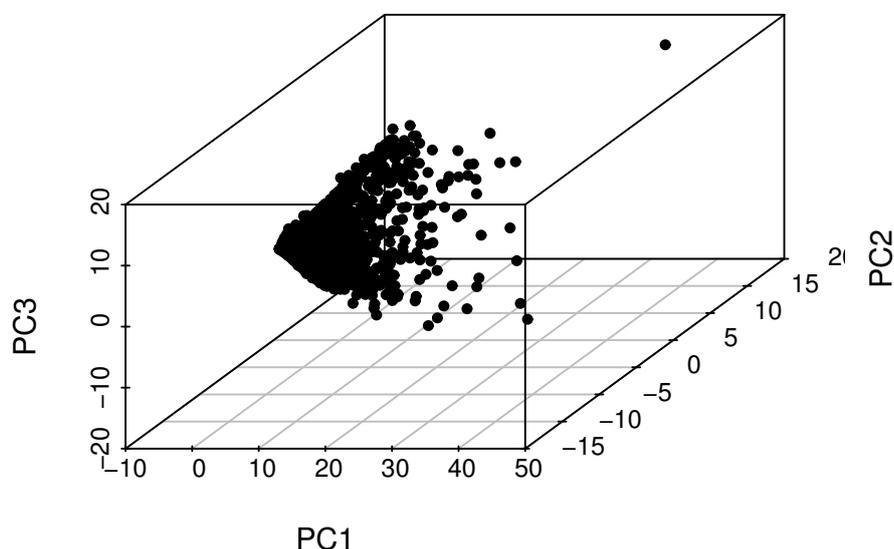


Figura 4.13 – Análise de agrupamento utilizando a técnica PCA que mostra a concentração da matriz em alguns alunos e objetos de aprendizagem. Os componentes são compostos das características *student_id*, *object_id* e *hours*. Foram normalizados para permitir a análise. Pode-se notar que o agrupamento dos *PC1*, *PC2* e *PC3* indicam que poucos alunos acessam poucos objetos com frequência.

A esparsidade não é a única razão para a imprecisão das recomendações fornecidas pela filtragem colaborativa. Se o sistema estiver lidando com preferências implícitas, as classificações geralmente são inferidas a partir das interações aluno/item como já mencionadas anteriormente no texto. Mapear preferências implícitas em classificações explícitas é uma tarefa não trivial e pode resultar em mapeamentos falsos. Isto é ainda mais verdadeiro para dados coletados em um servidor Web, pois os *logs* contêm informação limitada. Para determinar quanto tempo um aluno estava lendo um documento, deve-se calcular a diferença de data e hora de duas solicitações do mesmo. Isto, no entanto, não indica o tempo em que o aluno estava realmente lendo o documento ou se, por exemplo, saiu para almoçar, deixando o navegador aberto. Além disso, o usuário pode estar acessando informações em *cache* (a partir de um *cache* local ou de um *cache* intermediário do servidor *proxy*) não sendo possível detectar estes eventos no lado do servidor (WANG; WANG; YEUNG, 2015). Por exemplo, como mostrado na Figura 4.14, os tempos de acesso podem apresentar um certo ruído por limitações das requisições feitas via protocolo HTTP. Além disso, restrições como preferências de horário de acesso também devem ser consideradas e/ou incluídas para a recomendação de certos objetos de aprendizagem.

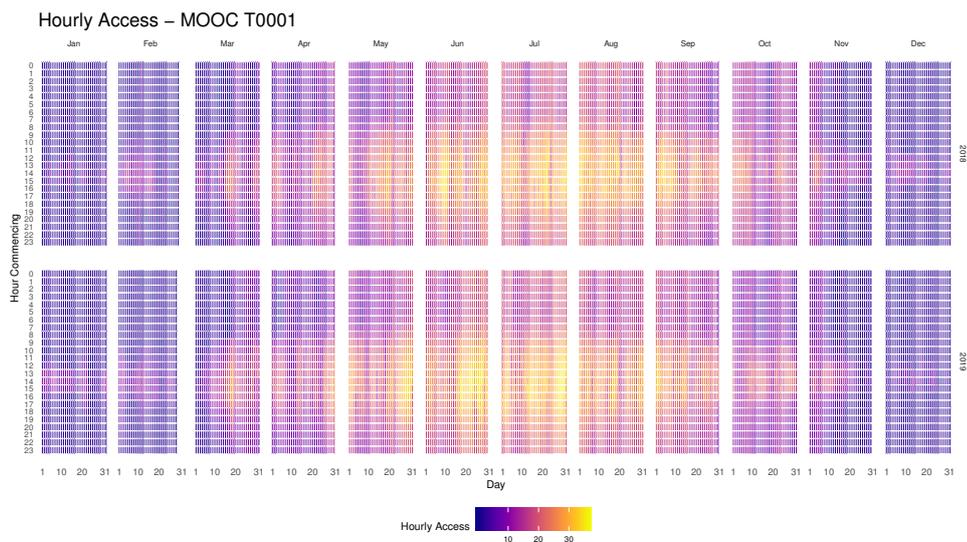


Figura 4.14 – Histórico das interações aluno/objeto coletado durante os anos de 2018 e 2019. Pode-se notar que o maior volume encontra-se durante os meses de março a novembro e os horários de acesso entre 08h00min e 19h00min.

Além disso, se um aluno não estiver logado e seu navegador não aceitar *cookies*, seria difícil rastreá-lo. Nestes casos, a única informação disponível que poderia potencialmente ajudar a rastrear o aluno é o seu endereço IP ³. No entanto, muitos alunos podem compartilhar o mesmo IP e, além disso, um usuário pode ter muitos endereços IP mesmo na mesma sessão. Os únicos mecanismos de rastreamento confiáveis são *cookies* e exigindo que os usuários efetuem *login* para acessar conteúdos relevantes.

A partir desta descrição dos problemas de dados, pode-se concluir que, para aplicar filtragem colaborativa, dados com baixa esparsidade são mais simples de serem processados para explicar e prever o comportamento do aluno. O pior cenário é ter dados altamente esparsos derivados de *logs* da Web. Neste caso, a motivação para filtragem colaborativa, baseada em *logs* da Web, tem a necessidade de uma abordagem mais apropriada, como as arquitetura de redes neurais conhecidas como *Deep Auto Encoder*.

4.2.3 Deep Auto Encoder

Os *Deep Auto Encoders* são redes de aprendizado que visam transformar entradas em saídas com a menor quantidade possível de distorção (WANG; WANG; YEUNG, 2015). Embora conceitualmente simples, eles desempenham um papel importante no aprendizado de máquina. Os *Deep Auto Encoders* foram introduzidos pela primeira vez na década de 1980 por Hinton e Grupo PDP para abordar o problema de aprendizagem não supervisionada, usando os dados de entrada como rótulos. Juntamente com as regras de aprendizado de Hebbian (WANG; WANG; YEUNG, 2015), os *Deep Auto Encoder*

³ O *Internet Protocol* (IP), ou Protocolo de Internet é um protocolo de comunicação usado entre todas as máquinas em rede para encaminhamento dos dados.

forneem um dos paradigmas fundamentais para o aprendizado não-supervisionados ou semi-supervisionado; a não necessidade de um conjunto de dados rotulados que possam ser utilizados em cenários de aprendizagem supervisionada. Conjunto de dados rotulados costumam ser difíceis de se obter em problemas reais. A Tabela 4.8 fornece uma visão de entrada de dados de um *Deep Auto Encoder*; uma matriz esparsa.

Tabela 4.8 – Exemplo de uma matriz esparsa. As células com ? representam a ausência de interações aluno/objeto

	object_id				
		1	2	...	n
student_id	1	1,10	?	...	?
	2	?	0,30	...	?

	n	?	?	?	1,00

Mais recentemente, os *Deep Auto Encoders* voltaram ao centro das atenções onde a arquitetura de rede neural é treinada com uma estratégia *feedforward* e de maneira não supervisionada, seguida por uma fase de aprendizado supervisionado para treinar a camada mais externa e ajustar toda a arquitetura. A fase *feedforward* é independente em relação à tarefa final e, portanto, pode ser usada em abordagens de transferência de aprendizado. Estas arquiteturas profundas (i.e. com várias camadas) demonstraram resultados avançados em vários problemas de classificação e regressão. Maiores detalhes podem ser obtidos na Seção 3.3. A Tabela 4.9 apresenta um possível resultado do processamento pelo *Deep Auto Encoder*.

Tabela 4.9 – Exemplo de uma matriz reconstruída por um *Deep Auto Encoder*. As células destacadas representam o valor previsto da interações aluno/objeto

	object_id				
		1	2	...	n
student_id	1	1,10	1,03	...	0,03
	2	0,90	0,30	...	0,09

	n	0,01	0,61	0,32	1,00

Portanto, uma arquitetura profunda deve ter n camadas de tamanho polinomial α , para algumas $\alpha > 0$ pequenas, em que n é o tamanho dos vetores de entrada. Mas este não é o caso nas arquiteturas descritas em (BALDI, 2012) e (KHAN; HAYAT; PORIKLI, 2019), que parecem ter profundidade constante ou, na melhor das hipóteses, logarítmica. A distinção entre profundidade finita e logarítmica torna-se quase impossível para os valores típicos de n usado em visão computacional, reconhecimento de fala e outros problemas típicos.

4.2.4 Hiperparâmetros

Hiperparâmetros são parâmetros configuráveis de um modelo. Pode ser a profundidade máxima de uma árvore de decisão, o número de vizinhos no algoritmo k -vizinhos mais próximos ou o número de camadas ocultas em uma rede neural. Em um sentido abstrato, é simplesmente um valor com um determinado intervalo de valores significativos, que pode ser ajustado para melhorar o desempenho do modelo. O objetivo da otimização do hiperparâmetro é encontrar a combinação dos hiperparâmetros de um modelo de maneira que maximize seu desempenho, enquanto investe uma quantidade sensata de recursos nele.

Uma maneira de encontrar um bom valor para um hiperparâmetro é começar determinando um intervalo de valores para ele. O intervalo muitas vezes pode ser deduzido do contexto razoavelmente bem. Em seguida, um processo pode amostrar a faixa uniformemente e usar o valor, o que permite ao modelo alcançar o melhor desempenho. Quando esta abordagem é usada para vários hiperparâmetros ao mesmo tempo, geralmente é chamada de pesquisa em grade. Para cada hiperparâmetro, escolhe-se uma lista de valores e treina-se o modelo com todas as combinações dos valores. Esta abordagem funciona bem para alguns hiperparâmetros, mas o número de modelos a serem treinados cresce rapidamente com o número de hiperparâmetros. Isso se torna especialmente problemático para os modelos, que possuem um grande número de hiperparâmetros e levam muito tempo para treinar. Isso acontece com as redes neurais. A Tabela 4.10 apresenta o conjunto de hiperparâmetros que podem ser utilizados para a construção de um modelo baseado na arquitetura de redes neurais *Deep Auto Encoder*.

Tabela 4.10 – Hiperparâmetros utilizados nos *Auto Encoder* (ORDWAY-WEST; PARVEEN; HENSLEE, 2018)

Hiperparâmetros	Significado	Valor
Função de ativação	Função que será utilização para ativar o neurônio	SELU
Tamanho do lote	Número de exemplos de treinamento usados em uma iteração	[1:Tamanho da Amostra]
<i>Dropout</i>	Taxa de neurônios desnecessários que serão descartados	[0:1]
Épocas	Número de iterações para o treinamento da rede	[1:∞]
<i>Factors</i>	Quantidade de neurônios em cada camada	[1:∞]
Camadas	Quantidade de camadas da rede	[1:∞]
Taxa de Aprendizagem	Taxa de atualização dos pesos da rede	[0:1]
<i>Weight Regularization</i>	Taxa para evitar <i>overfitting</i>	[0:1]

O número de modelos, que precisam ser treinados, para encontrar uma boa configuração de hiperparâmetros pode ser reduzido assumindo-se que existem dependências subjacentes, que podem ser modeladas. Usando esta suposição, um modelo do espaço de hiperparâmetros pode ser construído e usado para orientar o processo de amostragem, para que o processo treine principalmente modelos com valores de hiperparâmetros, que parecem promissores. Existem vários modelos que podem ser usados para este fim. Uma opção é a otimização *bayesiana* usando processos gaussianos (KHAN; HAYAT; PORIKLI, 2019), que parecem ter bons resultados empíricos.

4.2.5 Divisão do Conjunto de Dados

O conjunto de dados precisa ser dividido em conjuntos de treinamento, validação e teste. Isto simula efetivamente a situação, quando o modelo é treinado em dados fornecidos e posteriormente usado para previsões sobre novos dados. A avaliação de dados diferentes permite a verificação para descobrir se o modelo se generaliza bem. O conjunto de treinamento será usado para treinar o modelo. O conjunto de validação será usado para avaliar o modelo durante o treinamento para limitar o ajuste excessivo. O conjunto de testes será usado, quando o treinamento terminar, para gerar previsões e calcular métricas. Maiores detalhes sobre as estratégias para divisão do conjunto de dados podem ser obtidas na Seção 3.8.

O conjunto de treinamento conterá 3/5 do conjunto de dados original. Os conjuntos de validação e teste, cada um conterão 1/5. A divisão será ao longo da dimensão do usuário. Os itens permanecerão os mesmos em todos os subconjuntos. O processo de treinamento de uma rede neural é um processo iterativo. A estrutura da rede e o próprio processo de treinamento são governados por muitos hiperparâmetros. Encontrar bons valores para estes hiperparâmetros é uma parte importante desta tese. O conjunto de testes será usado para fins de avaliação. No entanto, antes de poder ser usado, ele precisa ser dividido em duas partes. Todos os usuários e todos os itens estarão presentes nas duas partes. Para calcular muitas das métricas descritas na Seção 2.2, dois conjuntos de classificações precisam ser criados. Uma matriz (a matriz de previsão) conterá a maioria das classificações e a segunda matriz (a matriz retida) conterá o restante.

Esta divisão permite gerar previsões para classificações usando a matriz de previsão e compará-las com as classificações reais na matriz retida. Isso simula a situação em que um aluno recebe uma recomendação. Suas preferências reais (conhecidas do conjunto retido) são comparadas às preferências previstas dos modelos. A matriz realizada conterá 20% das interações aluno/objeto. Como durante a divisão, alguns alunos na matriz de previsão podem ficar sem interações, estes alunos serão removidos. Como resultado, as duas matrizes têm as mesmas dimensões, mas alguns alunos foram removidos em comparação com a matriz original.

4.2.6 Métricas de Avaliação

O problema de recomendação pode ser visto como uma generalização de regressão da seguinte maneira. Na regressão, os dados estão na forma de uma matriz com colunas representando recursos e uma coluna, que representa a variável de destino. O algoritmo prevê os valores da variável de destino com base nos valores dos recursos. Na recomendação (especialmente na filtragem colaborativa), os dados são uma matriz com colunas representando itens. É igualmente válido inverter a matriz e fazer com que as colunas representem usuários. O problema permanece o mesmo. Não há distinções entre

as colunas. O objetivo é preencher os valores ausentes em qualquer coluna. Devido a esta conexão, muitas das métricas básicas de avaliação da regressão podem ser aplicadas. Por exemplo, um erro quadrático médio pode ser calculado. No entanto, como o objetivo típico é a lista principal de recomendações, estas métricas não correspondem muito bem às expectativas do sistema de recomendação.

4.2.6.1 Online Evaluation

Nesta perspectiva, a melhor maneira de avaliar um sistema de recomendação é o teste *A/B*. O comerciante começa com o serviço já em funcionamento, conecta parte dos alunos ao novo sistema e compara os resultados com o antigo sistema de recomendação. A métrica de avaliação pode ser qualquer coisa que tenha um valor educacional, por exemplo, a taxa de acesso de alunos que clicam nos objetos de aprendizagem, que foram recomendados. No entanto, isso nem sempre é uma opção e pode ser feito apenas com um sistema que já é relativamente estável. Portanto, as métricas de avaliação *off-line* também são necessárias. Criar uma métrica *off-line* com correlação com os resultados de testes *on-line* é uma tarefa difícil. Para poder realizar uma avaliação *off-line*, é necessário um conjunto de dados históricos de interações do aluno. Como foi discutido anteriormente, existem objetivos diferentes para o sistema de recomendação. Por este motivo, várias métricas devem ser calculadas para tentar atingir objetivos diferentes. Alguns dos mais comuns são os seguintes.

4.2.6.2 Accuracy

Devido à semelhança com a regressão, a *accuracy* pode ser aplicada com muita facilidade. Como parte do objetivo dos sistemas de recomendação é prever as preferências do usuário, a precisão pode ser usada para medir este aspecto. Um exemplo de uma função popular nesta categoria é a raiz do erro quadrático médio (RMSE).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{P}_i - P_i)^2}{n}} \quad (4.24)$$

A Equação 4.24 representa como calcular o erro quadrático médio para n valores preditos. \hat{P}_i representa o valor predito e P_i o valor original.

4.2.6.3 Recall

O *Recall* é uma métrica que se concentra na lista principal de itens recomendados para um determinado usuário u . O cálculo é realizado usando um conjunto de itens retido, no qual o usuário clicou. A lista *top-k* recomendada gerada com base nas classificações

restantes é comparada com o conjunto retido. Quanto mais itens da lista *top* – *k* puderem ser encontrados no conjunto, mais próximo será o valor resultante da recuperação.

$$Recall@K(u, \omega) = \frac{\sum_{k=1}^K \mathbb{1}[\omega(k) \in I_u]}{\min(K, |I_u|)} \quad (4.25)$$

A Equação 4.25 mostra como calcular a chamada para a lista de itens *K*. A função $w(k)$ representa um item na posição *k* na lista *top*–*k* e a função *I* em uma função indicadora, que determina se o item está presente no conjunto retido. Finalmente, o denominador é o mínimo de *K* e o número de itens clicados pelo usuário *u*. Isso normaliza o valor para variar entre zero e um.

4.2.6.4 Normalized Discounted Cumulative Gain

NDCG é uma métrica semelhante à recuperação. Ela se concentra na lista principal de itens recomendados e a compara ao conjunto de itens retidos. Ao contrário do *recall*, ele também considera a ordem dos itens na lista *top*–*k*. Quando um item em melhor posição é encontrado no conjunto retido, ele contribui com um peso maior. O peso é dado pela função monotonicamente decrescente (por exemplo, logarítmico), como mostrado na Equação 4.26.

$$NDCG@K(u, \omega) = \sum_{k=1}^K \frac{2^{\mathbb{1}[\omega(k) \in I_u]} - 1}{\log(K + 1)} \quad (4.26)$$

NDCG@*K* é o DCG@*K* normalizado linearmente para $[0, 1]$ após a divisão pelo melhor DCG@*K* possível, onde todos os itens ocultos são classificados no topo.

4.2.6.5 Coverage

A cobertura do catálogo (CC) é uma medida da fração do banco de dados de itens, recomendada pelo sistema para pelo menos um usuário. Conforme discutido nas seções anteriores, alguns sistemas podem ter problemas com a recomendação de novos itens. Como um dos objetivos dos sistemas de recomendação é a diversidade, a fração de itens já recomendados é relevante.

$$CC = \frac{\cup_{u=1}^m T_u}{n} \quad (4.27)$$

A Equação 4.27 mostra como calcular a cobertura do catálogo para o banco de dados de *n* itens e a lista *top*–*k* gerada para *m* usuários. T_u representa a lista *top* – *k* do usuário *u*.

4.2.6.6 Mean Average Precision

Esta métrica identifica o quanto a lista contendo os k objetos de aprendizagem, para cada aluno, é relevante para todos os alunos do conjunto de dados. Tal métrica é definida pela Equação 4.28.

$$MAP@N = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{1}{M} \sum_{k=1}^N P_u(k) \times rel_u(k) \quad (4.28)$$

A métrica da Equação 4.28 é utilizada para medir a acurácia do sistema de recomendação e pode ser utilizada para medir o quanto cada modelo gerado é mais eficaz se comparado aos demais.

4.2.7 Treinamento da Rede

Para explicar como as redes neurais podem ser treinadas, primeiro uma função de perda deve ser definida. A função de perda (*loss function*) representa um erro entre a saída da rede neural e a saída esperada no conjunto de dados de treinamento.

$$Loss = |y - h_w(x)|^2 = \sum_{k=1}^n (y_k - a_k)^2 \quad (4.29)$$

A Equação 4.29 é um exemplo de erro quadrático médio usado como função de perda. O vetor y representa o vetor de saída esperado e $h_w(x)$ representa a rede neural com a_k sendo um elemento da saída. Portanto, o erro está entre a saída esperada e a real da rede. O processo de treinamento é uma otimização usando descida de gradiente. Para poder usar este método, toda a função de perda precisa ser diferenciável em relação aos pesos \mathbf{w} .

$$\frac{\partial}{\partial w} Loss = \frac{\partial}{\partial w} \sum_{k=1}^n (y_k - a_k)^2 = \sum_{k=1}^n \frac{\partial}{\partial w} (y_k - a_k)^2 \quad (4.30)$$

Para os pesos de saída na camada de saída, os gradientes podem ser derivados de forma relativamente simples, como mostrado na Figura 4.8. No entanto, toda a rede pode ser uma função complexa. A saída de cada camada é a entrada da próxima e, como resultado, cada camada é uma função aninhada extra. No entanto, toda a função pode ser decomposta usando a regra da cadeia. Os pesos são atualizados camada por camada, começando pela camada de saída e retornando pela rede. É por isso que o processo é chamado de retropropagação de erro.

$$w_{i,j} \leftarrow w_{i,j} + \alpha \times \alpha_i \times \Delta_j \quad (4.31)$$

Cada peso pode ser atualizado pela Equação 4.31. O valor do peso $w_{i,j}$ da conexão entre os neurônios i e j muda entre as iterações. A mudança é dada pela taxa de aprendizado α , pela saída α_i do neurônio i, j pelo delta propagado Δ_j . O valor de Δ_j é determinado pela soma dos deltas Δ_k de neurônios na próxima camada, ponderada pelos pesos $w_{j,k}$. O último elemento importante do cálculo é a derivada da função de ativação (Equação 4.32).

$$\Delta_j = g'(in_j) \sum_{k=1}^n w_{j,k} \Delta_k \quad (4.32)$$

Desta forma, a utilização das métricas de avaliação devem ser feitas em conjunto para possam auxiliar na construção do modelo computacional que seja mais adequado as recomendações de objetos de aprendizagem aos aluno. Além disso, estas métricas devem auxiliar na configuração dos hiperparâmetros a fim de se obter a melhor combinação. Portanto, além de permitir entender a acurácia do modelo, as métricas permite a otimização dos hiperparâmetros da rede neural artificial.

5 Resultados e Análises

Neste capítulo os resultados dos experimentos relacionados com a tarefa de classificação e recomendação são apresentados, analisados e discutidos. Os detalhes para a interpretação dos resultados são evidenciados fazendo uma ligação entre a teoria, implementação e saídas geradas, como modelos, métricas e gráficos. Além disso, os hiperparâmetros para a construção são apresentados para permitir a reprodução e/ou comparação com demais estratégias de otimização de hiperparâmetros e/ou construção de modelos a fim de solucionar a mesma classe de problemas. Por fim, os resultados gerados são apresentados de forma gráfica para uma melhor compreensão.

5.1 Classificação

Neste experimento um conjunto de dados rotulados foi utilizado para o treinamento de uma rede neural artificial baseada na arquitetura *Multi Layer Perceptron* (MLP). O aprendizado supervisionado foi empregado com a divisão do conjunto de dados feito via *Leave-One-Out Cross Validation* (LOOCV), função de perda *Root Mean Square Error* (RMSE) e métricas de avaliação baseadas na matriz de confusão. Todos os conceitos relacionados podem ser vistos, em detalhes, no Capítulo 3.

5.1.1 Conjunto de Dados de Treinamento

Nesta seção, os resultados das experiências são apresentados e discutidos. Inicialmente, investigaram-se os aspectos de três tipos de variáveis: (i) descritores de contagem, (ii) descritores de tempo e (iii) descritores de destino onde, este último é do tipo contagem. Não foram encontrados *outliers* no conjunto de dados. A mediana dos descritores de contagem de tipo era de cerca de 4 acessos por elemento (*content_visit*, *outline_visit*, etc.), como mostra a Figura 5.2. Os descritores de tempo definem o tempo gasto em cada elemento (*content_stay*, *outline_stay*, etc.). O valor zero (0) representa que o elemento não teve acesso. O tempo médio gasto no tempo de digitação é de cerca de 60 segundos e há uma restrição que limita o acesso em 120 segundos, pois este é o limite da de tempo da sessão. Finalmente, a mediana das variáveis-alvo é de cerca de 0, que expressa o conjunto de dados do estilo de aprendizagem equilibrado em cada dimensão (entrada, processamento, percepção e entendimento), significando que os alunos estão distribuídos igualmente entre as classes ativa e reflexiva (GOMEDE; BARROS; MENDES, 2020). Estes valores são mostrados, representados de forma proporcional à quantidade de alunos estudados, na Figura 5.1.

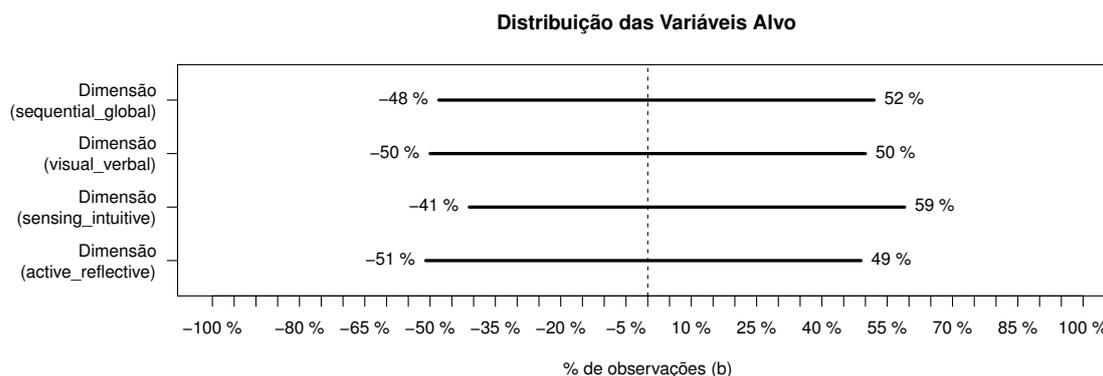


Figura 5.1 – A distribuição das observações do conjunto de dados de cada dimensão. O % das preferências de cada aluno é representada como 48% sequenciais, 52% globais, 50% visuais e 50% verbais, 41% intuitivos e 59% reflexivos, 51% ativos e 49% reflexivos.

Como mostrado na Figura 5.2, os valores representam o comportamento dos alunos interagindo com os objetos de aprendizagem disponíveis no LMS. Os comportamentos dos descritores do tipo tempo e contador são apresentados.

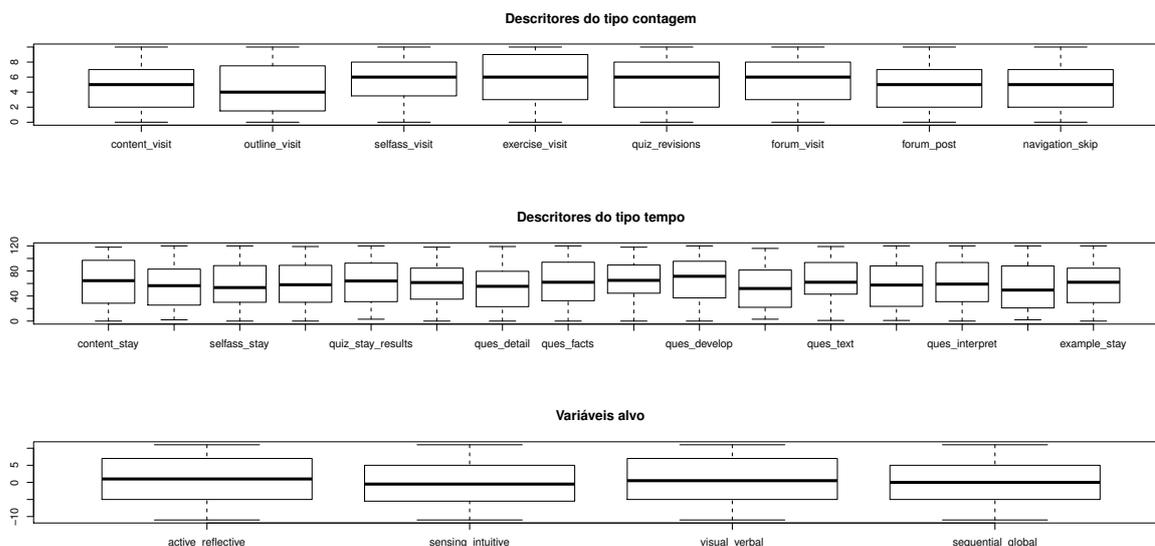


Figura 5.2 – A análise de *boxplot* do conjunto de dados (descritores e metas de contagem e tempo). Há uma limitação para o descritor de tempo em 120 e a contagem em 10 vezes.

Também explorou-se a frequência de cada dimensão de preferência antes da transformação de destino em variáveis binárias (Equações 4.2 à 4.5). Como resultado, obteve-se os estilos de aprendizagem dos alunos para cada dimensão. As dimensões *active_reflective*, *sensing_intuitive* e *sequential_global* apresentaram uma distribuição aproximadamente uniforme; no entanto, a dimensão *visual_verbal* apresentou uma concentração próxima de -5 , o que representa uma preferência dos alunos em adquirir informações

visuais. A Figura 5.3 mostra esta análise.

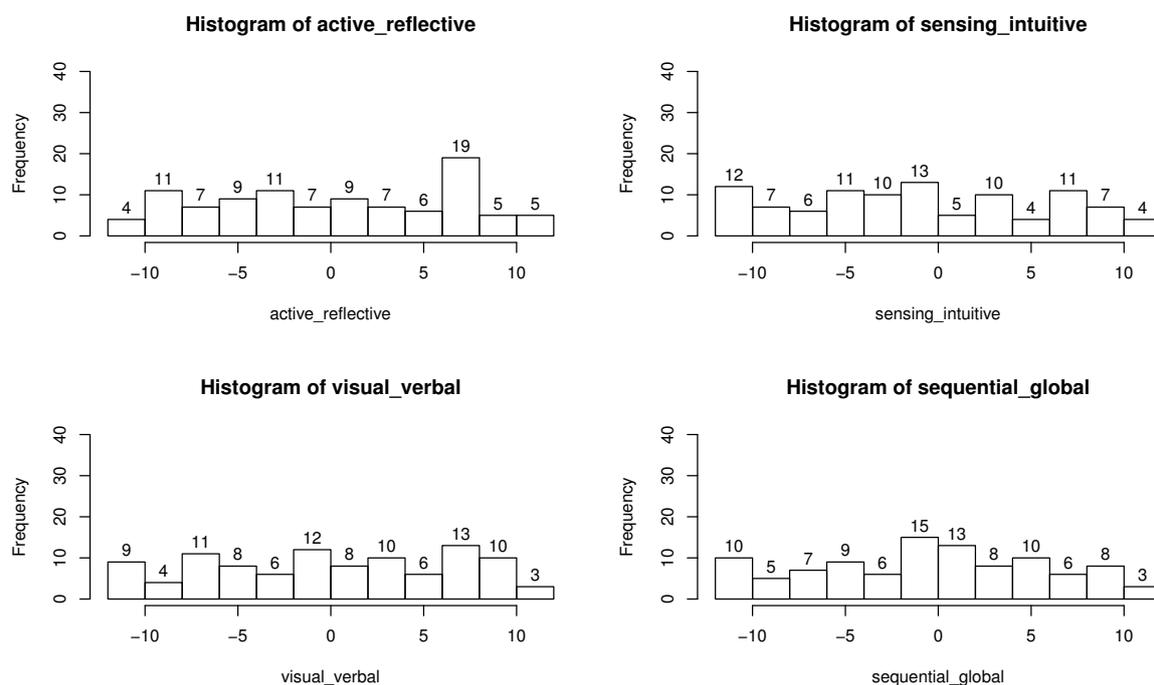


Figura 5.3 – Distribuição da frequência do estilo de aprendizagem de cada dimensão. Tal distribuição aparentemente não mostra nenhum padrão.

Para a dimensão *active_reflective* pode-se notar que existe certa tendência para a polaridade reflexiva. Já para a dimensão *sensitive_intuitive*, pode-se notar que existe certa tendência para a polaridade sensitiva. Para a dimensão *visual_verbal* pode-se notar que existe certa homogeneidade para as polaridades. Por fim, para a dimensão *sequential_global* pode-se notar que existe certa distribuição normal.

5.1.2 Agrupamentos

Também investigou-se a possibilidade de usar o conjunto de dados para identificar a preferência do aluno. Se, por exemplo, um determinado conjunto de atributos representar uma das quatro dimensões de aprendizado, estes atributos podem ajudar na redução da dimensionalidade e melhorar a precisão do classificador (BERNARD et al., 2017). Os grupos foram investigados usando o algoritmo *k-means* para identificar grupos naturais no conjunto de dados. O algoritmo de médias *k* foi usado com $k = 2, 3, 4, 5$. Como resultado, foram obtidos agrupamentos com 2 e 3 grupos com baixa sobreposição. Usando um número de grupos de até 3, os *clusters* resultantes se sobrepuseram. A sobreposição de grupos indica que os dados não podem ser separados de forma linear¹. Estes resultados são mostrados na Figura 5.4.

¹ A separação linear é provida pelo algoritmo de rede neural conhecido como *perceptron*.

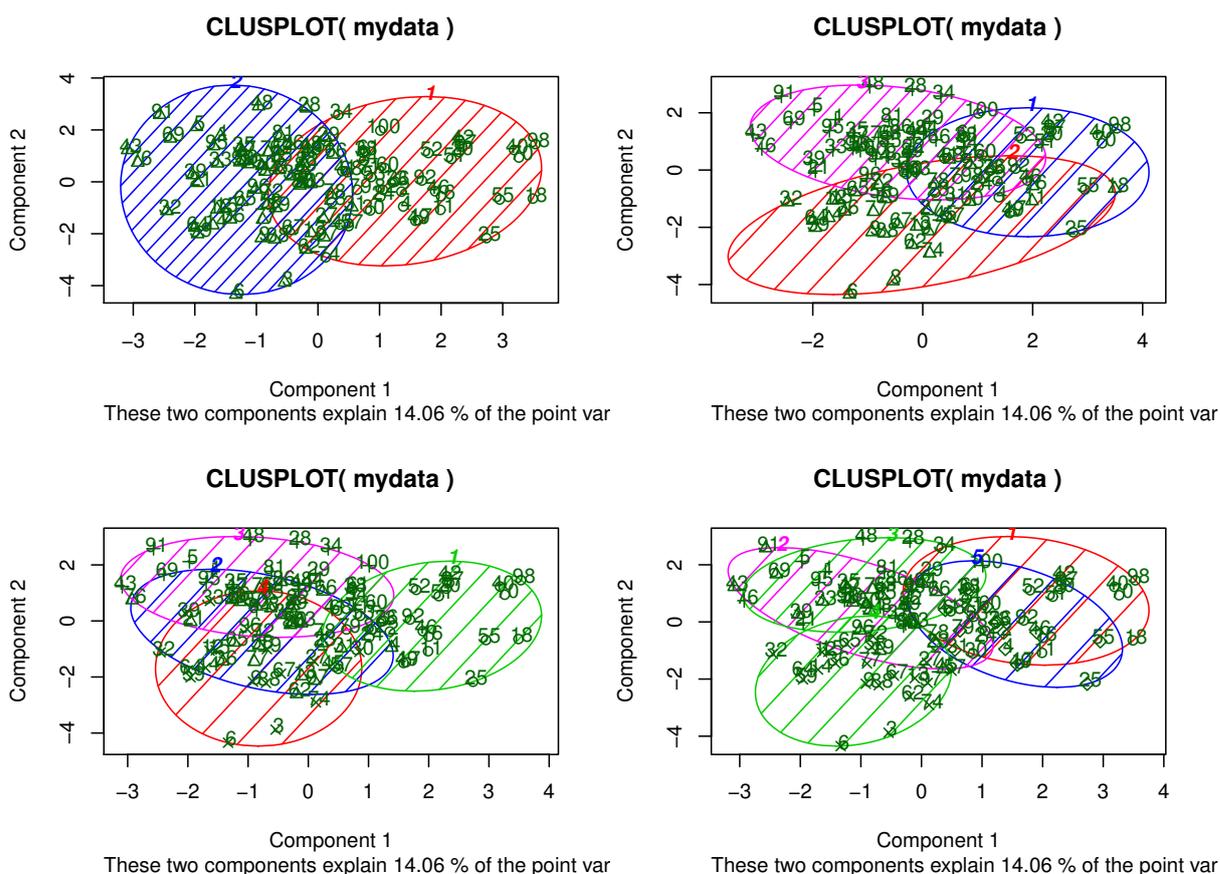


Figura 5.4 – *Clusters* naturais no conjunto de dados usando médias de k (com $k = 2, 3, 4, 5$). Nenhum *cluster* natural foi identificado e, portanto, as classes não são lineares separadas (HEIDARI et al., 2020).

Além disso, outra técnica de análise, a Análise de Componentes Principais, ou *Principal Component Analysis* (PCA), foi aplicada para investigar outros atributos ou correlações relevantes e se os alvos, de cada dimensão, podem ser explicados por algum descritor (contagem e tempo). Esta é uma questão importante para a redução da dimensionalidade, a fim de melhorar a precisão e reduzir o custo de construção do modelo. Utilizando esta técnica investigou-se a possibilidade de alguns dos descritores serem mais expressivos do que outros para determinar o resultado das classes. Entretanto, tal fenômeno não foi observado no conjunto de dados, sendo, portanto, que nenhum dos descritores tem uma relevância sensível maior que a dos demais. Desta forma, optou-se por utilizar todos os descritores no classificador, mesmo que isto represente um aumento no custo computacional. Existe um problema ao se utilizar muitas características para modelos computacionais conhecido pelo nome *curse of dimensionality*². Estes resultados são mostrados na Figura 5.5.

² A maldição da dimensionalidade se refere a vários fenômenos que surgem ao analisar e organizar dados em espaços de alta dimensão que não ocorrem em configurações de baixa dimensão, como o espaço físico tridimensional da experiência cotidiana. A expressão foi cunhada por Richard E. Bellman ao considerar problemas em programação dinâmica.

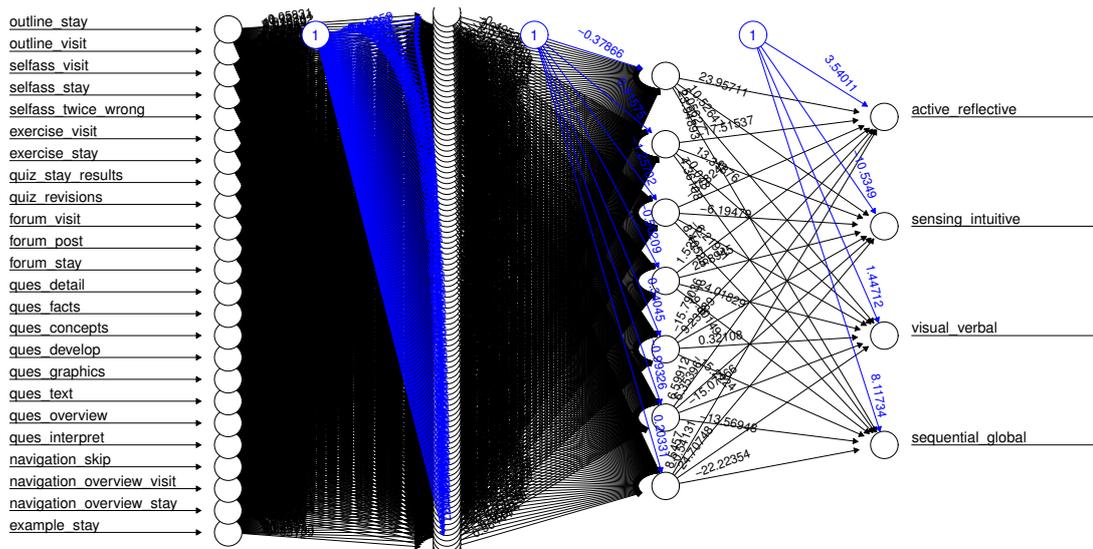


Figura 5.6 – O modelo construído com 2 camadas ocultas, 26 neurônios de entrada e 4 neurônios de saída. Os pesos representados em cada sinapse (conexão) representam o conhecimento induzido pelo treinamento da rede.

5.1.4 Métricas

Este modelo avalia todos os descritores, fornecendo simultaneamente a classificação dos alunos em cada dimensão de aprendizado. Este é um algoritmo de destino com várias previsões (TRUONG, 2016). Todas as equações foram usadas para avaliar o modelo. Utilizou-se matriz de confusão para classificar as previsões. Os resultados, de cada dimensão, são mostrados na Tabela 5.1.

Tabela 5.1 – Métricas e respectivos resultados para cada dimensão

Processing (active_reflective)	Perception (sensing_intuitive)
Accuracy : 0.85	Accuracy : 0.76
95% CI : (0.2306, 0.6847)	95% CI : (0.3605, 0.8088)
No Information Rate : 0.6	No Information Rate : 0.55
P-Value [Acc >NIR] : 0.9435	P-Value [Acc >NIR] : 0.4143
Kappa : -0.0784	Kappa : 0.1919
Mcnemar's Test P-Value : 0.5465	Mcnemar's Test P-Value : 1.0000
Sensitivity : 0.5000	Sensitivity : 0.5556
Specificity : 0.4167	Specificity : 0.6364
Pos Pred Value : 0.3636	Pos Pred Value : 0.5556
Neg Pred Value : 0.5556	Neg Pred Value : 0.6364
Prevalence : 0.4000	Prevalence : 0.4500
Detection Rate : 0.2000	Detection Rate : 0.2500
Detection Prevalence : 0.5500	Detection Prevalence : 0.4500
Balanced Accuracy : 0.4583	Balanced Accuracy : 0.5960
'Positive' Class : 1	'Positive' Class : 1
Input (visual_verbal)	Understanding (sequential_global)
Accuracy : 0.75	Accuracy : 0.80
95% CI : (0.4078, 0.8461)	95% CI : (0.1912, 0.6395)
No Information Rate : 0.65	No Information Rate : 0.65
P-Value [Acc >NIR] : 0.601	P-Value [Acc >NIR] : 0.9940
Kappa : 0.2045	Kappa : -0.2371
Mcnemar's Test P-Value : 1.000	Mcnemar's Test P-Value : 0.7728
Sensitivity : 0.7692	Sensitivity : 0.4615
Specificity : 0.4286	Specificity : 0.2857
Pos Pred Value : 0.7143	Pos Pred Value : 0.5455
Neg Pred Value : 0.5000	Neg Pred Value : 0.2222
Prevalence : 0.6500	Prevalence : 0.6500
Detection Rate : 0.5000	Detection Rate : 0.3000
Detection Prevalence : 0.7000	Detection Prevalence : 0.5500
Balanced Accuracy : 0.5989	Balanced Accuracy : 0.3736
'Positive' Class : 1	'Positive' Class : 1

O melhor modelo construído alcançou a precisão de 85%, 76%, 75% e 80% em cada atributo de destino. Além disso, utilizou-se a matriz de confusão para calcular a Curva Característica de Operação do Receptor, ou *Receiver Operating Characteristic Curve* (ROC) e a Área Sob a Curva, *Area Under the Curve* (AUC) (HEIDARI et al., 2020). Para cada alvo, os resultados foram superiores a Bernard et. al (BERNARD et al., 2017). A Figura 5.9 mostra estes resultados.

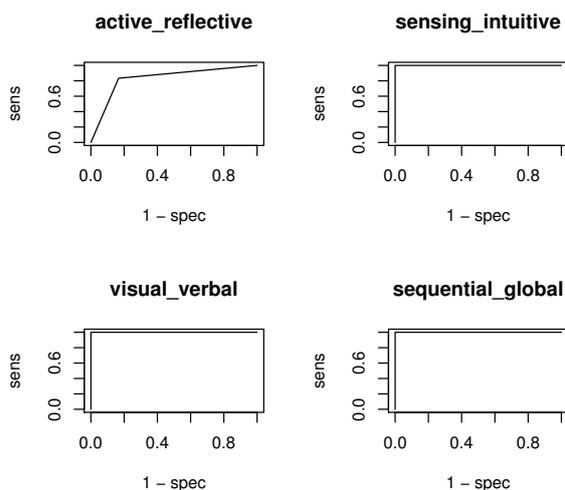


Figura 5.7 – Modelo

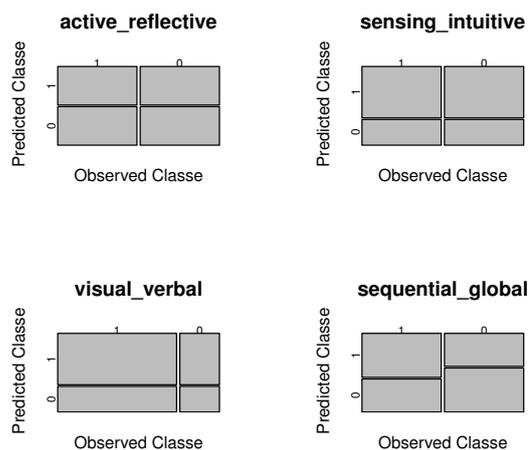


Figura 5.8 – Treinamento

Figura 5.9 – A análise ROC de cada dimensão da matriz de confusão que ilustra a capacidade de diagnóstico do sistema classificador binário, pois seu limiar de discriminação é variado. (a) Representa a matriz de confusão individual e (b) representa a curva ROC individual.

Todas as métricas indicaram que o modelo pode ser um método para classificar automaticamente um aluno em um LMS. A relação entre os descritores melhora a precisão (conforme mostrado nas métricas **especificidade** e **sensibilidade** da Tabela 5.1). Além disso, a previsão de múltiplos alvos (MTP) preocupa-se com a previsão simultânea de múltiplas variáveis-alvo de diversos tipos, e o modelo que utiliza o procedimento Felder-Silverman é de longe a teoria mais popular aplicada aos sistemas de *e-learning* (NORMADHI et al., 2019). Enquanto isso, de outro ponto de vista, a precisão dos resultados usando a abordagem proposta poderia ser melhorada quando executada em um ambiente de grande conjunto de dados. Outra limitação do estudo atual é que os resultados dos experimentos foram testados apenas em uma plataforma com um assunto específico. A consistência do desempenho precisa ser testada quando executada com diferentes plataformas de gerenciamento de aprendizado ou outros cursos *on-line*.

5.2 Recomendação

Neste experimento 3 tipos de arquiteturas de *Deep Auto Encoders* foram utilizados (*Collaborative Denoising Auto Encoders*, *Deep Auto Encoder for Collaborative Filtering* e *Deep Auto Encoder for Collaborative Filtering With Content Information*) além de um modelo baseado em popularidade para comparação dos resultados. Um conjunto de dados contendo a interação de alunos com objetos de aprendizagem foi utilizado para a reconstrução da matriz esparsa com os valores mais prováveis para cada interação aluno/objeto.

5.2.1 Divisão do Conjunto de Dados

O conjunto de dados apresentado na Seção 4.2.1 foi utilizado para a divisão em dois subconjuntos, um de treinamento e um de testes, utilizando o algoritmo **Random Sampling** conforme explicado na Seção 3.8.2. Para isso, 80% do conjunto de dados foi destinado ao treinamento da rede neural artificial e 20% foi destinado para os testes.

5.2.2 Popularity Model

Este modelo faz recomendações apresentando os objetos de aprendizagem mais populares; em outras palavras, aqueles que tiveram mais visualizações (em horas) durante um período. Tal recomendação não é adaptada sendo a mesma para todos os alunos. Este modelo base será usado como referência para os demais gerados pelos *Deep Auto Encoders*. Este modelo representa a soma, em ordem decrescente, do tempo gasto com cada um dos objetos. Portanto, tem-se o primeiro objeto o mais popular entre os alunos.

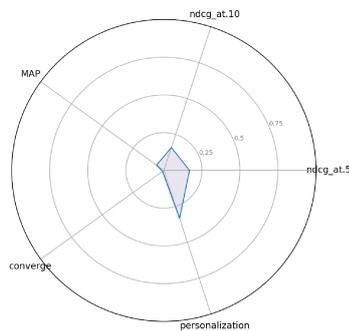


Figura 5.10 – Mean Average Precision

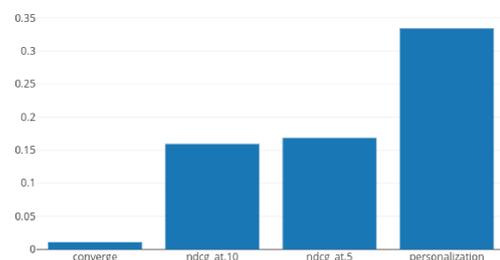


Figura 5.11 – Métricas de Personalização

Figura 5.12 – Resultado com as métricas do modelo de popularidade. Este modelo não fornece personalização mas é utilizado como base para a comparação dos demais modelos gerados. Pode-se notar que a métrica *personalization* apresentou o maior valor relativo (a) e (b).

Pode-se notar, na Figura 5.12, que o valor alcançado para a métrica de personalização foi de 0,334, estando em uma escala de [0 : 1]. Em outras palavras, este valor representa 33% de personalização e não é muito útil para Aprendizado Adaptativo. Mas pode ser uma saída para o problema de *cold-start* para novos alunos (KIRAN; KUMAR; BHASKER, 2020).

5.2.3 Deep Auto Encoder

Esta arquitetura de rede neural artificial reconstrói os dados de entrada na saída da rede. Isso permite com que, no caso de uma matriz esparsa, os dados faltantes possam ser criados de maneira que os mesmos sejam uma predição adequada. Para maiores detalhes consulte a Seção 3.2.

5.2.3.1 Collaborative Denoising Auto Encoders

O *Collaborative Denoising Auto Encoders* (CDAE) aprende representações distribuídas dos alunos e objetos por meio da formulação dos dados de *feedback* dos objetos do aluno usando uma estrutura *Denoising Auto Encoder* (WU et al., 2016). A CDAE assume que, seja qual forem as interações aluno/objeto observadas são uma versão corrompida do conjunto completo de preferências do aluno. O modelo aprende representações latentes de preferências de objetos de alunos corrompidos que podem reconstruir melhor a entrada de dados (WU et al., 2016). Tais hiperparâmetros são mostrados na Tabela 5.2.

Tabela 5.2 – *Collaborative Denoising Auto Encoders* - Hiperparâmetros

Name	Value
activation	selu
batch	64
dropout	0.8
epochs	50
factors	500
layers	[128,256,128]
lr	0.0001
name	cdae
reg	0.0001

A *Scaled Exponential Linear Unit* (SELU) é uma função de ativação das mais recentes. Ao usar esta função de ativação na prática, é necessário usar *lecun_normal* para a inicialização dos pesos. Para ser aplicado, deve-se usar o *Alpha Dropout* sendo representado por dois valores; um alfa α e lambda λ de acordo com a Equação 5.1.

$$\alpha \approx 1.6732632423543772848170429916717 \quad (5.1)$$

$$\lambda \approx 1.0507009873554804934193349852946 \quad (5.2)$$

Eles têm muitos números após o ponto decimal para precisão absoluta e são predeterminados, o que significa que não é necessário escolher o valor α correto para esta função de ativação. Todas as funções de ativação mais recentes parecem uma combinação das outras funções de ativação existentes. A equação para isso é (Equação 5.3):

$$SELU(x) = \begin{cases} x\lambda & \text{if } x > 0 \\ \alpha e^x - \alpha & \text{if } x \leq 0 \end{cases} \quad (5.3)$$

Ou seja, se o valor de entrada x é maior que zero, o valor de saída se torna x multiplicado por lambda λ . Se o valor de entrada x é menor ou igual a zero, tem-se uma função que vai até ∞ , que é a nossa saída y , quando x é zero. Essencialmente, quando x é menor que zero, multiplica-se α pelo exponencial do valor x menos o valor α e multiplica-se pelo valor λ .

O *droupout* permite remover neurônios da rede que estão gerando *overfitting* do modelo. O valor otimizado é próximo de 0,8 (SRIVASTAVA et al., 2014). O parâmetro *epoch* refere-se a quantidade de vezes que as observações serão apresentadas para a rede durante o período de treinamento. O valor 50 está em torno de 10% acima do valor necessário para a rede convergir, ou seja, não apresentar mais melhoras na taxa de erro da *loss function*. O *factors* representa o tamanho dos fatores que serão utilizados para a criação do espaço latente. O hiperparâmetro *layers* representa quais serão as quantidades de neurônios e cada uma das camadas ocultas. Por exemplo, [100, 10, 100] representa uma rede com 3 camadas ocultas sendo que a primeira possui 100 neurônios, a segunda 10 e a terceira 100. O parâmetro *lr* representa a taxa de aprendizagem (*learning rate*). Maiores detalhes podem ser vistos na Seção 3.4. O hiperparâmetro *reg* refere-se a outra técnica para evitar *overfittig* chamada de *Weight Regularization* (KHAN; HAYAT; PORIKLI, 2019). Estes parâmetros foram avaliados utilizando as estratégias de otimização discutidas anteriormente. Obter uma configuração ótima (ou seja, a melhor combinação de hiperparâmetros possível) é uma tarefa complexa. Portanto, uma busca heurística foi realizada a fim de obter uma boa solução a um custo possível. Com os hiperparâmetros ajustados o modelo CDAE foi criado utilizando os conceitos de *Deep Auto Encoder* apresentados na Seção 4.2. Além disso, utilizou-se a entrada de dados normalizada para a rede neural artificial. Para isto, uma técnica de *Feature Scaling* foi utilizada ³. O resultado do treinamento, com as métricas de avaliação, pode ser visto na Figura 5.15.

³ O dimensionamento de recursos (*Feature Scaling*) no aprendizado de máquina é uma das etapas mais críticas durante o pré-processamento de dados antes de criar um modelo de aprendizado de máquina. O escalonamento pode fazer a diferença entre um modelo de aprendizado de máquina fraco e um melhor. As técnicas mais comuns de dimensionamento de recursos são Normalização e Padronização. A normalização é usada quando queremos limitar nossos valores entre dois números, normalmente, entre [0, 1] ou [-1, 1]. Enquanto a padronização transforma os dados em uma média zero e uma variância de 1, eles tornam nossos dados sem unidade. A preparação de dados envolve o uso de técnicas como a normalização e padronização para redimensionar as variáveis de entrada e saída antes de treinar um modelo de rede neural.

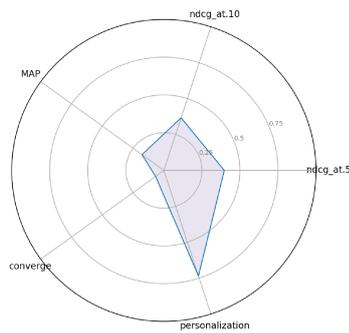


Figura 5.13 – Mean Average Precision

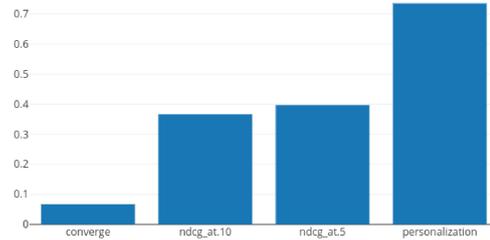


Figura 5.14 – Métricas de Personalização

Figura 5.15 – Resultado com as métricas do *Collaborative Denoising Auto-Encoders*. Este modelo fornece uma boa personalização e é utilizado para a comparação dos demais modelos gerados. Pode-se notar que a métrica *personalization* apresentou o maior valor relativo (a) e (b).

Como se pode notar, ao comparar os resultados do método *Popularity Model* com o CDAE, houve uma melhora significativa em todas as métricas, principalmente na de personalização, que foi melhorada em torno de 50%. O modelo gerado e o histórico de treinamento podem ser observados na Figura 5.18.

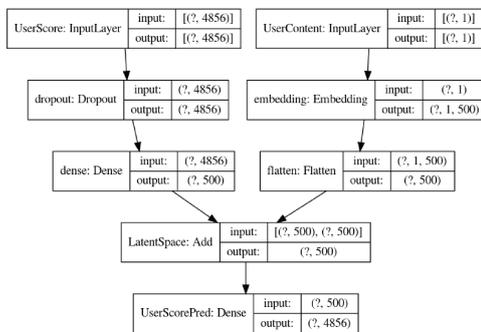


Figura 5.16 – Modelo

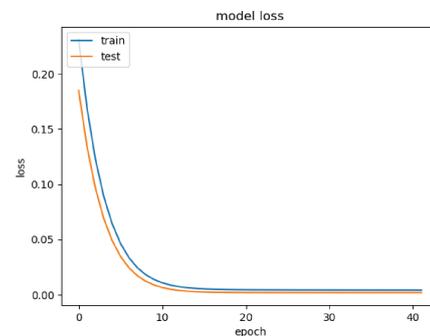


Figura 5.17 – Treinamento

Figura 5.18 – Modelo treinado utilizando o conjunto de dados particionados. Este modelo possui as camadas de entrada e saída de dados (ambos do mesmo tamanho) e as camadas ocultas com as matrizes fatoradas. Histórico do treinamento da rede. Pode-se notar o mesmo comportamento do treinamento da rede MLP. A taxa de erro é mais alta no início, uma rápida redução e a convergência da rede.

A convergência da rede aconteceu entre a 10 e a 20 época. A *loss function* para o treinamento e para o teste tiveram o mesmo comportamento. Por fim, pode-se observar os 10 primeiros objetos de aprendizagem que foram recomendados para o aluno 25. Esta escolha do usuário foi aleatória para demonstrar como a lista de objetos a serem recomendados pode ser construída. Tal resultado, com a lista de objetos a serem

recomendados para o aluno 25 são apresentados na Tabela 5.3.

Tabela 5.3 – Lista dos top@10 gerados por *Collaborative Denoising Auto Encoders*

	score	object_id	learning_object
0	0.737318	4572	Learning Object 52
1	0.677891	3509	Learning Object 418
2	0.633031	1262	Learning Object 214
3	0.619094	501	Learning Object 145
4	0.612064	2	Learning Object 10
5	0.592862	3188	Learning Object 389
6	0.588873	2631	Learning Object 338
7	0.579694	4705	Learning Object 64
8	0.565154	4601	Learning Object 546
9	0.557259	4509	Learning Object 51

5.2.3.2 Deep Auto Encoder for Collaborative Filtering

Um *Deep Auto Encoder* tem a tendência de generalizar melhor do que os *Auto Encoders* com poucas camadas. Além disso, as funções de ativação não lineares com partes negativas, são importantes para o treinamento do modelo e o uso de técnicas de regularização, como o *dropout*, são necessárias para evitar o *overfitting* (KUCHAIEV; GINSBURG, 2017). Os parâmetros utilizados para a criação do modelo podem ser vistos na Tabela 5.4.

Tabela 5.4 – *Deep Auto Encoder for Collaborative Filtering* - Hiperparâmetros

Name	Value
activation	selu
batch	64
dropout	0.8
epochs	50
factors	15
layers	[512,256,512]
lr	0.0001
name	auto_enc
reg	0.01

O *Deep Auto Encoder* apresentou uma melhora de aproximadamente 10% com relação ao CDAE. Isto se deve a existência de camadas intermediárias, que podem lidar melhor com a esparsidade da matrizes ao resolver um problema não linear (HEIDARI et al., 2020). Além disso, as camadas intermediárias permitem uma maior acurácia do modelo para o problema (WANG; WANG; YEUNG, 2015). As métricas resultantes podem ser vistas na Figura 5.21.

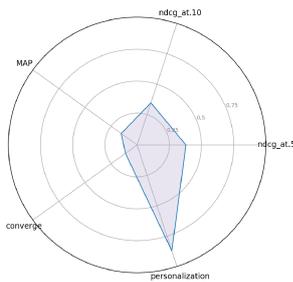


Figura 5.19 – Mean Average Precision

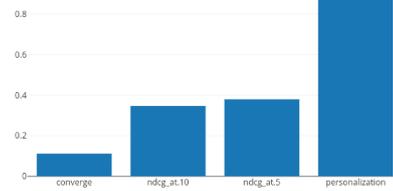


Figura 5.20 – Métricas de Personalização

Figura 5.21 – Resultado com as métricas do *Deep Auto Encoder for Collaborative Filtering*. Este modelo fornece uma boa personalização e é utilizado para a comparação dos demais modelos gerados. Pode-se notar que a métrica *personalization* apresentou o maior valor relativo (a) e (b).

De forma complementar, a Figura 5.24 apresenta o modelo gerado pelo treinamento da rede e o histórico de treinamento. Pode-se notar que a convergência foi próxima do CDAE, estando entre 10 e 20 épocas.

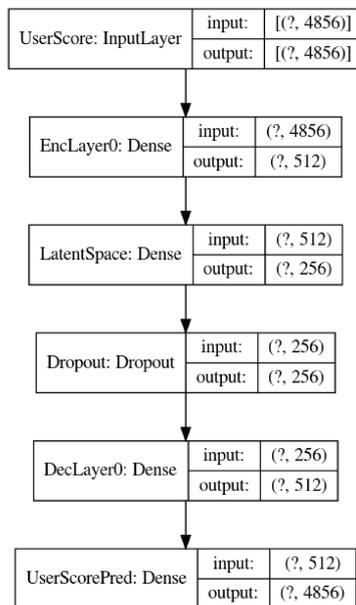


Figura 5.22 – Modelo

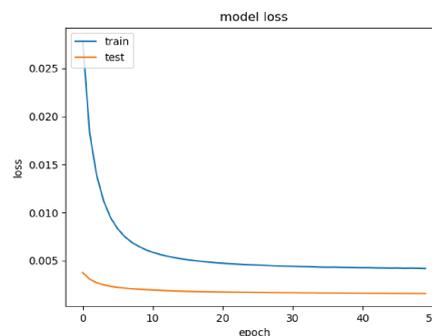


Figura 5.23 – Treinamento

Figura 5.24 – (a) Modelo treinado utilizando o conjunto de dados particionados. Este modelo possui as camadas de entrada e saída de dados (ambos do mesmo tamanho) e as camadas ocultas com as matrizes fatoradas. (b) Histórico do treinamento da rede. Pode-se notar o mesmo comportamento do treinamento da rede MLP. A taxa de erro é mais alta no início, uma rápida redução e a convergência da rede.

A Figura 5.24 contem o modelo gerado com 1 camada de entrada, sendo que cada objeto da matriz esparsa representa um neurônio. Uma camada de espaço latente

que transforma a camada anterior de 4.000 neurônios para 512. Esta é uma camada oculta. Depois disso, tem-se mais uma camada oculta que transforma os 512 neurônios em 256. A reconstrução é feita nas demais camadas até chegar na camada de saída. Por fim, pode-se observar na Tabela 5.5 os objetos de aprendizagem recomendados para o aluno 25.

Tabela 5.5 – *Deep Auto Encoder for Collaborative Filtering*

	score	object_id	learning_object
0	0.842951	3188	Learning Object 389
1	0.828034	2	Learning Object 10
2	0.794385	4675	Learning Object 612
3	0.764501	4572	Learning Object 52
4	0.728598	1328	Learning Object 22
5	0.683430	501	Learning Object 145
6	0.606181	1262	Learning Object 214
7	0.583904	1992	Learning Object 28
8	0.568082	972	Learning Object 188
9	0.552525	445	Learning Object 14

5.2.3.3 Deep Auto Encoder for Collaborative Filtering With Content Information

Este modelo é uma adaptação do modelo apresentado anteriormente, mas adicionando informações de conteúdo. Desta forma, o modelo é uma implementação híbrida. Neste modelo, o *nome do objeto* de todos os objetos de aprendizagem que o aluno já interagiu foi incluído como informações adicionais para a filtragem colaborativa. Esta é uma maneira de adicionar informações de conteúdo ao nível do usuário (KUCHAIEV; GINSBURG, 2017). Os hiperparâmetros para a construção deste modelo podem ser vistos na Tabela 5.6.

Tabela 5.6 – *Deep Auto Encoder for Collaborative Filtering With Content Information*

Name	Value
activation	selu
batch	64
dropout	0.8
epochs	50
factors	15
layers	[512,256,512]
lr	0.0001
name	auto_enc_content
reg	0.01

O *Deep Auto Encoder With Content Information* apresentou uma melhora de aproximadamente 5% com relação ao *Deep Auto Encoder*. Isso se deve a existência de camadas intermediárias que podem lidar melhor com a esparsidade da matriz (WANG; WANG; YEUNG, 2015). As métricas resultantes podem ser vistas na Figura 5.27.

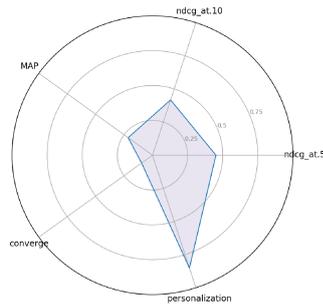


Figura 5.25 – Mean Average Precision

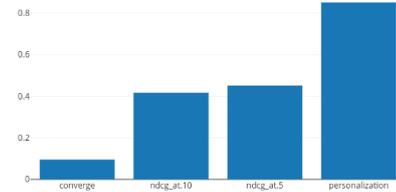


Figura 5.26 – Métricas de Personalização

Figura 5.27 – Resultado com as métricas do *Deep Auto Encoder With Content Information*. Este modelo fornece uma boa personalização e é utilizado para a comparação dos demais modelos gerados. Pode-se notar que a métrica *personalization* apresentou o maior valor relativo (a) e (b).

Adicionalmente, a Figura 5.30 apresenta o modelo gerado pelo treinamento da rede e o histórico de treinamento, estando a convergência próxima do CDAE, entre 10 e 20 épocas.

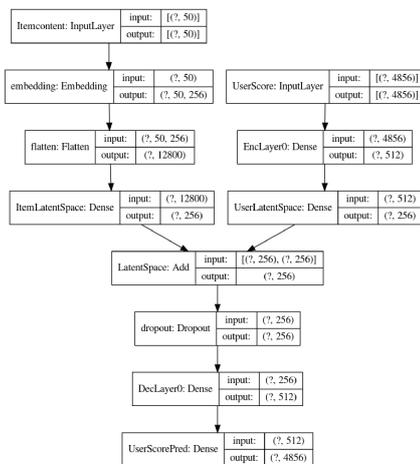


Figura 5.28 – Modelo

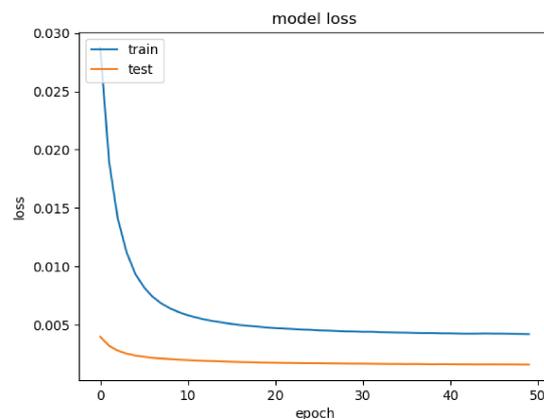


Figura 5.29 – Treinamento

Figura 5.30 – Modelo treinado utilizando o conjunto de dados particionados. (a) Este modelo possui as camadas de entrada e saída de dados (ambos do mesmo tamanho) e as camadas ocultas com as matrizes fatoradas. (b) Histórico do treinamento da rede. Pode-se notar o mesmo comportamento do treinamento da rede MLP. A taxa de erro é mais alta no início, uma rápida redução e a convergência da rede.

A Figura 5.24 contem o modelo gerado com 1 camada de entrada, sendo que cada objeto da matriz esparsa representa um neurônio. Uma camada de espaço latente que transforma a camada anterior de 4.000 neurônios para 512. Esta é uma camada oculta. Depois disso, tem-se mais uma camada oculta que transforma os 512 neurônios em 256. A

reconstrução é feita nas demais camadas até chegar na camada de saída. Por fim, pode-se observar na Tabela 5.7 os objetos de aprendizagem recomendados para o aluno 25.

Tabela 5.7 – *Deep Auto Encoder for Collaborative Filtering With Content Information*

	score	object_id	learning_object
0	1.278175	4401	Learning Object 5
1	1.191606	555	Learning Object 150
2	1.040922	4594	Learning Object 54
3	0.927504	4994	Learning Object 90
4	0.918778	4391	Learning Object 499
5	0.897038	202	Learning Object 118
6	0.888085	4738	Learning Object 67
7	0.886164	4402	Learning Object 50
8	0.865007	136	Learning Object 112
9	0.858167	4885	Learning Object 801

5.2.4 Discussão

Cada um dos quatro métodos apresentados podem gerar uma lista de recomendações de objetos de aprendizagem para um aluno. O método com melhores resultados foi o *Deep Auto Encoder*. Grande parte destes resultados vem da quantidade de camadas ocultas que foram adicionadas o que permite uma melhor reconstrução da saída da matriz esparsa.



Figura 5.31 – Distribuição

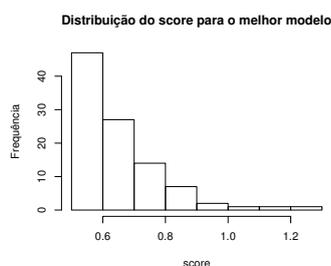


Figura 5.32 – Frequência

Figura 5.33 – A distribuição do *score* gerado apresenta alguns valores como *outliers*. Estes valores contribuem positivamente para métricas como $\text{top}@N$ (onde N é o número de objetos recomendados) para N com valores menores (por exemplo, $N = 10$). Em (a) pode-se notar o comportamento logarítmico de decaimento. Em (b) pode-se notar a frequência dos valores de *score*.

A Figura 5.33 mostra a distribuição de *score* por objetos que segue a mesma distribuição de cauda longa. A frequência de distribuição mostrar que muitos objetos tem um *score* baixo, ou seja, não são tão relevantes para o aluno e que poucos objetos tem score altos, sendo que estes devem ser os primeiros a serem recomendados. Pode-se notar,

também, que os primeiros objetos a serem recomendados apresentam-se como *outliers* se distanciando dos demais. Isto pode ser observado na Figura 5.34.

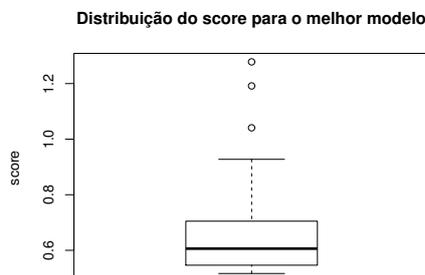


Figura 5.34 – Os primeiros objetos a serem recomendados apresentam-se como *outliers* se distanciando dos demais.

Em termos comparativos, podem-se utilizar as métricas fornecidas pelos autores (BOBADILLA; SERRADILLA; HERNANDO, 2009; WANG; WANG; YEUNG, 2015; KRAUSS, 2016; PROTASIEWICZ et al., 2016; BARBIERI et al., 2017; NAFEA; SIEWE; HE, 2019; MEDIO et al., 2020). Pode-se notar (PROTASIEWICZ et al., 2016) e (BARBIERI et al., 2017) foram os que apresentaram resultados mais expressivos. Adicionalmente, esta pesquisa complementa estes trabalhos apresentando uma melhora significativa com relação a tais autores. Isto pode ser visto na Figura 5.35.

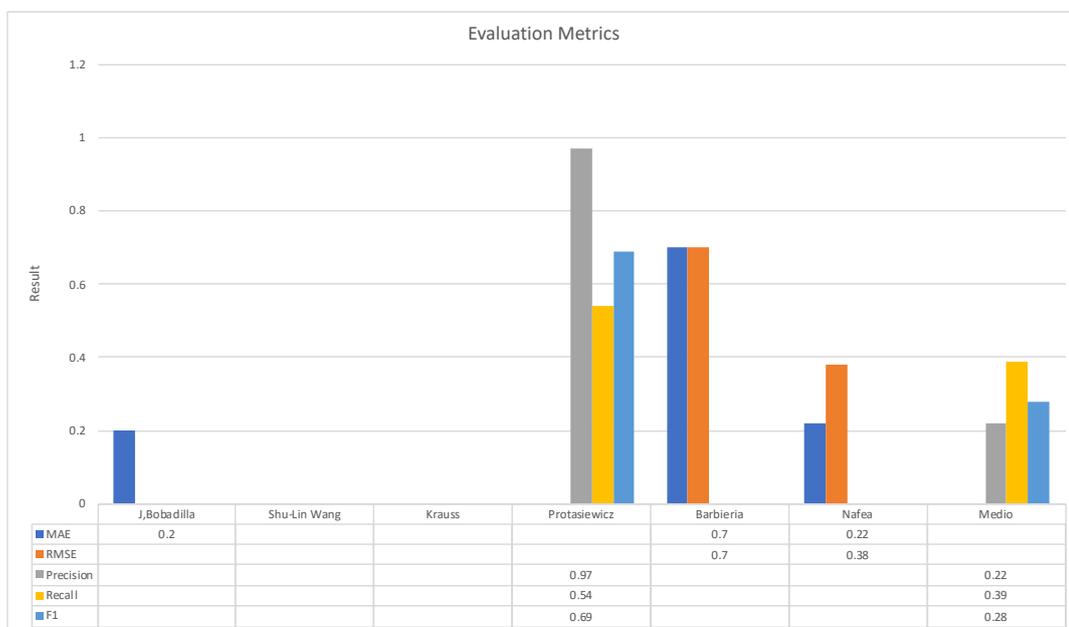


Figura 5.35 – A comparação dos resultados, para cada métrica disponível, dos trabalhos publicados. Pode-se notar que dois trabalhos apresentaram resultados mais expressivos. Entretanto, nenhum deles apresentou um conjunto de métricas mais abrangente.

SCORM, pode-se utilizar diversos **metadados** para complementar o *Deep Auto Encoder for Collaborative Filtering With Content Information* com as informações destes objetos. Isto pode melhorar as métricas de recomendação e, adicionalmente, melhorar o nível de aprendizagem dos alunos. Um exemplo destes metadados pode ser visto a seguir.

```
1 {
2     "user_id": "1",
3     "name": "Everton Gomedes",
4     "created_at": "27/07/2020",
5     "updated_at": "27/07/2020",
6     "age": "23",
7     "country": "BR",
8     "city": "Campinas"
9 }
10
11 {
12     "item_id": "1",
13     "name": "Auto Encoder loss function",
14     "created_at": "23/01/2010",
15     "updated_at": "24/04/1982",
16     "category": ["Visual", "Global"],
17     "tags": ["Loss function", "Deep learning"],
18     "lesson_id": "1",
19     "module_id": "1",
20     "course_id": "1",
21     "last_played_date": "20/06/1979",
22     "autor_id": "2"
23 }
```

Estes metadados podem ser obtidos quando um objeto de aprendizagem for utilizado por um aluno. De forma adicional, podem-se empregar tais metadados para fornecer dimensões adicionais a matriz esparsa com as interações aluno/objeto.

6 Conclusão

6.1 Conclusões Gerais

Esta pesquisa tem como objetivo investigar o uso de algoritmos de inteligência computacional (IC), conhecidos como redes neurais artificiais, para analisar e melhorar a precisão do uso de abordagens autônomas de Aprendizado Adaptativo em *e-learning*. A hipótese de trabalho pode ser descrita como: se o padrão de aprendizagem de um aluno puder ser reconhecido então pode-se descrever e prever seu comportamento em um LMS. Portanto, o padrão de aprendizagem do aluno está relacionado com a forma e o conteúdo que este tem maior predisposição à aprendizagem. O Aprendizado Adaptativo pode ser definido como a adaptação do formato e/ou conteúdo visando melhorar o desempenho de um aluno ao utilizar um LMS. Deve-se notar que a utilização do LMS deve-se ao fator eficiência. No entanto, a eficácia de tal método pode ser alcançada mesmo sem o suporte de um LMS. As arquiteturas de redes utilizadas para a inferência dos modelos foram *Multi Layer Perceptron*, *Collaborative Denoising Auto Encoders*, *Deep Auto Encoder for Collaborative Filtering* e *Deep Auto Encoder for Collaborative Filtering With Content Information*.

Para a identificação das preferências com o **formato**, um experimento, utilizando dados de 100 alunos foi realizado, demonstrando a precisão geral da abordagem para cada uma das quatro dimensões do estilo de aprendizagem *Felder-Silverman Learning Model Model* (FSLSM). Pode-se observar a partir dos resultados obtidos que esta abordagem pode ser usada para identificar o estilo de aprendizagem dos alunos com base em seu comportamento nos MOOC. Esta abordagem reduz o ruído dos questionários (BERNARD *et al.*, 2017; NORMADHI *et al.*, 2019), permite a classificação quando necessário para verificar se o estilo mudou ao longo do tempo (WILLINGHAM, 2019) e permite que estes dados sejam armazenados para uso futuro. Assim, ao identificar os estilos de aprendizagem dos alunos, os sistemas adaptativos de aprendizagem podem usar estas informações para fornecer um Aprendizado Adaptativo mais preciso, levando a uma maior satisfação e redução do tempo de aprendizado (BARBIERI *et al.*, 2017). Além disso, os alunos podem se beneficiar diretamente da identificação mais precisa dos estilos de aprendizagem, sendo capazes de alavancar seus pontos fortes em relação aos estilos de aprendizagem e entender suas fraquezas. Adicionalmente, os professores podem usar estas informações de estilo de aprendizagem para fornecer aos alunos conselhos mais precisos, o que é novamente mais útil para os alunos quanto à identificação mais precisa do estilo de aprendizagem. De forma complementar, outras partes interessadas no ecossistema educacional, como pais, professores e administradores, podem fazer uso desta abordagem para melhorar a educação

em geral (FELDER; SILVERMAN et al., 1988). Para a identificação das preferências com o **conteúdo**, 4 experimentos foram realizados, por meio das arquiteturas de redes neurais artificiais *Auto Encoder*. Esta proposta traz uma maior granularidade de aprendizado adaptativo permitindo a criação de uma assinatura de preferência do aluno com relação aos objetos de aprendizagem. Ambas as abordagens mostraram-se válidas e podem ser utilizadas em conjunto, a primeira para identificar o perfil de um aluno e a segunda para recomendar objetos de aprendizagem para o mesmo. Além das contribuições para a aplicação de redes neurais artificiais, com as arquiteturas de *multi layer perceptron* e *auto encoder*, este trabalho apresenta resultados práticos que podem melhorar a qualidade da educação, não somente para alunos, mas para todos os envolvidos neste ambiente (GOMEDE et al., 2018).

Para o modelo de classificação as métricas que utilizadas são a *F1* e *Accuracy*. A primeira apresenta uma visão geral do classificador (média harmônica de falsos positivos e negativos) e seus resultados foram 87%, 91%, 93%, 85% para cada dimensão sendo, portanto, considerado um bom resultado. A segunda apresenta o desempenho geral do classificador e, como os dados não estão desbalanceados, pode ser usada. O resultados de tal métrica foram de 85%, 76%, 75%, 80% para cada dimensão sendo novamente um bom resultado. Comparado com Bernard (BERNARD et al., 2017) o modelo gerado apresenta uma acurácia superior além de tratar dos atributos e rótulos simultaneamente (GOMEDE; BARROS; MENDES, 2020). Já para os modelos de recomendação utilizou-se as métricas MAP e NDCG discutidas na Seção 4.2.6. A métrica MAP apresenta a média dos k objetos de aprendizagem recomendados para todos os alunos. Desta forma, pode-se avaliar o comportamento geral do recomendador. A métrica NDCG é complementar a MAP utilizando, adicionalmente, a posição do objeto de aprendizagem na lista de k objetos. Esta métrica considera que quanto mais próximo da posição 0 da lista de objetos, mais este seria recomendado. Ou seja, tal métrica permite avaliar ser um objeto, dentro da lista de k objetos, é mais relevante que os demais. De forma similar, a métrica avalia a lista dos k objetos recomendados de forma média para todos os usuários. Neste caso, os modelos gerados pela arquitetura *Deep Auto Encoder for Collaborative Filtering With Content Information* apresentaram um valor mais relevante para as métricas MAP e NDCG sendo, 23% e 45%, respectivamente. Além disso, o recomendador está sujeito às restrições do ambiente educacional discutidas na Seção 2.1. Em comparação com demais autores, como (BOBADILLA; SERRADILLA; HERNANDO, 2009; WANG; WU, 2011; KRAUSS, 2016; PROTASIEWICZ et al., 2016; BARBIERI et al., 2017; NAFEA; SIEWE; HE, 2019; MEDIO et al., 2020), os modelos gerados são melhores para a métrica RMSE, vista na Equação 4.24 (em torno de 83%, comparado ao trabalho de (BARBIERI et al., 2017) com 70%, conforme visto na Figura 5.35) além de utilizar arquiteturas consideradas como o estado da arte (*Deep Auto Encoder*) e permitir a utilização do padrão SCORM.

Tais métricas do classificador (*F1* e *Accuracy*) indicam que as preferências do

aluno podem ser identificadas, com uma certa margem de erro, e que tal padrão pode ser utilizado para prever o comportamento do aluno e fornecer os conteúdos em um formato adequado. A margem de erro é oriunda da estratégia de rotulação, ruídos nos dados, estratégias de divisão do conjunto de dados e métricas de avaliação. Apesar disso, os modelos gerados podem ser úteis ao contexto de *e-learning*. As métricas de recomendação corroboram a hipótese pois apresentam que a lista de objetos recomendados, com relação ao conteúdo, são da preferência dos alunos. Portanto, o padrão de preferência, relacionado ao conteúdo, também pode ser identificado. Além disso, esta maneira de identificação é mais granular e pode ser utilizada para identificação em um nível individual. Tal identificação é apresentada nesta tese como uma assinatura de preferências. Tal assinatura também contém uma certa margem de erro pelos mesmos motivos citados anteriormente (exceto pela rotulação que não é necessária a arquitetura *Deep Auto Encoder*), mas o modelo pode ser utilizado para prever as preferências dos alunos com relação ao conteúdo. De forma geral, os modelos de classificação podem ser utilizados para identificação das preferências dos alunos para receber os conteúdos e os modelos de recomendação a preferência para receber conteúdos relacionados. Tais modelos permitem a adaptação requerida em sistemas de aprendizado adaptativo e corroboram a hipótese de, uma vez identificado o padrão de aprendizagem, seu comportamento em um LMS pode ser previsto.

6.2 Implicações Práticas

Dentre as implicações práticas deste trabalho, pode-se citar a possibilidade de identificar os estilos de aprendizagem de um aluno. Isso pode auxiliar os demais agentes do sistema educacional a planejar melhor os conteúdos e métodos de ensino visando aumentar o desempenho de seus alunos. Além disso, a estratégia mais granular de personalização, permite a recomendação de objetos de aprendizagem que podem transformar o sistema de *e-learning* em algo adaptado exclusivamente à preferências do aluno. De forma complementar, existe a possibilidade de agrupar alunos de várias maneiras, por similaridade ou dissimilaridade. Podem-se recomendar, além de objetos de aprendizagem, professores, alunos, cursos, pesquisas e tudo mais que possa ser traduzido para a matriz usuário/item. Por fim, pode-se criar, de forma mais ampla, a sequência de aprendizagem dos alunos em longo prazo (GOMEDE et al., 2018), uma vez identificadas suas preferências e que estas possam ser reavaliadas para auxiliar os envolvidos no processo de aprendizagem de forma mais ampla. Por fim, pode-se utilizar a inteligência computacional em outros cenários, como determinação do desempenho do aluno de acordo com suas características. Isto pode auxiliar a encontrar os pontos que podem ser melhorados e trabalhados (GOMEDE et al., 2018).

6.3 Trabalhos Futuros

Sugestões para trabalhos futuros incluem a aplicação prática desta abordagem por meio de *plug-ins* para os LSM, como o Moodle. Diferentes algoritmos podem ser testados comparando seus resultados com os trabalhos da rede neural artificial, uma vez que este trabalho apresenta valores de referência baseados na *Confusion Matrix* e métricas atuais de sistemas de recomendação (MAP e NDCG) que podem ser replicados para outros algoritmos. Questões sociais também podem ser investigadas para identificar se elas influenciam os estilos de aprendizagem. Pode-se como os estilos de aprendizagem funcionam na propagação de informações em redes baseadas em redes complexas. Este é um tópico importante, com grande impacto em aplicativos do mundo real, porque é uma base para os sistemas de recomendação e pode ser usado para melhorar o processo de aprendizado dos alunos. Adicionalmente, dentre as possibilidades de trabalhos futuros, uma delas envolve recomendações aos demais agentes como professores, administradores, pesquisadores e pais e/ou responsáveis (GOMEDE et al., 2018). Isto permite com que toda a interação entre os agentes possam ser guiadas pelo sistema adaptativo de *e-learning*.

Adicionamento, a utilização de diversos repositórios de objetos de aprendizagem para fornecer uma maior amplitude de aprendizagem (MEDIO et al., 2020). Além disso, pode realizar recomendações para além de objetos como a recomendação de professores, colegas, departamentos, caminhos de aprendizagem e tudo mais que possa ser convertido em uma matriz usuário/item. Adicionalmente, pode-se expandir a matriz bi-dimensional aluno/objeto para demais dimensões como o tempo, a localização, entre outras (KRAUSS, 2016). De forma adicional, podem-se empregar os metadados (fornecido pelos objetos de aprendizagem) para fornecer um nível maior de personalização. Estes metadados são padronizados pelo SCORM e entregues via repositórios de objetos de aprendizagem. Como exemplo de tais metadados, pode-se citar o nome do objeto, a data de criação, o assunto a qual ele pertence, quantidade de tempo para execução, tipo, autores, área de concentração, entre outros. De forma adicional, podem-se utilizar os objetos de aprendizagem, padronizados pelo SCORM, para permitir um maior gerenciamento por parte dos professores, tutores e administradores. O que permite uma forma de entender como os alunos aprendem do ponto de vista de conteúdo, para adaptar a continuidade ou não dos objetos. Por exemplo, um objeto que, mesmo sendo ofertado, apresenta um nível baixo de rendimento pode ser descartado sendo que outro mais eficiente pode entrar em seu lugar. Por fim, pode-se estudar a possibilidade de Aprendizado Adaptativo para outros estilos de aprendizagem que venham a surgir ou que possam ser atualizados.

Referências

- ALRAIMI, K. M. Hangjung zo, andrew p. ciganek, understanding the moocs continuance. *Computers & Education*, v. 80, p. 28–38, 2015. Citado 2 vezes nas páginas 32 e 83.
- AROYO, L. et al. Interoperability in personalized adaptive learning. *Journal of Educational Technology & Society*, JSTOR, v. 9, n. 2, p. 4–18, 2006. Citado na página 26.
- BA-OMAR, H.; PETROUNIAS, I.; ANWAR, F. A framework for using web usage mining to personalise e-learning. In: IEEE. *Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)*. [S.l.], 2007. p. 937–938. Citado na página 46.
- BALDI, P. Autoencoders, unsupervised learning, and deep architectures. In: *Proceedings of ICML workshop on unsupervised and transfer learning*. [S.l.: s.n.], 2012. p. 37–49. Citado 2 vezes nas páginas 58 e 98.
- BARAK, M.; WATTED, A.; HAICK, H. Motivation to learn in massive open online courses: Examining aspects of language and social engagement. *Computers & Education*, Elsevier, v. 94, p. 49–60, 2016. Citado na página 31.
- BARBIERI, J. et al. Autoencoders and recommender systems: Cofils approach. *Expert Systems with Applications*, Elsevier, v. 89, p. 81–90, 2017. Citado 10 vezes nas páginas 29, 30, 39, 43, 52, 53, 58, 121, 124 e 125.
- BERNARD, J. et al. Learning style identifier: Improving the precision of learning style identification through computational intelligence algorithms. *Expert Systems with Applications*, Elsevier, v. 75, p. 94–108, 2017. Citado 22 vezes nas páginas 15, 27, 28, 29, 34, 39, 42, 46, 48, 49, 50, 74, 75, 77, 78, 79, 80, 82, 107, 111, 124 e 125.
- BOBADILLA, J.; SERRADILLA, F.; HERNANDO, A. Collaborative filtering adapted to recommender systems of e-learning. *Knowledge-Based Systems*, Elsevier B.V., v. 22, n. 4, p. 261–265, 2009. ISSN 09507051. Disponível em: <<http://dx.doi.org/10.1016/j.knosys.2009.01.008>>. Citado 5 vezes nas páginas 29, 52, 53, 121 e 125.
- BODILY, R.; VERBERT, K. Review of research on student-facing learning analytics dashboards and educational recommender systems. *IEEE Transactions on Learning Technologies*, v. 10, n. 4, p. 405–418, 2017. ISSN 19391382. Citado 2 vezes nas páginas 31 e 33.
- BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In: *Proceedings of COMPSTAT'2010*. [S.l.]: Springer, 2010. p. 177–186. Citado 3 vezes nas páginas 35, 60 e 61.
- BRUSILOVSKY, P.; KARAGIANNIDIS, C.; SAMPSON, D. Layered evaluation of adaptive learning systems. *International Journal of Continuing Engineering Education and Life Long Learning*, Inderscience Publishers, v. 14, n. 4-5, p. 402–421, 2004. Citado na página 30.

BUDER, J.; SCHWIND, C. Learning with personalized recommender systems: A psychological view. *Computers in Human Behavior*, v. 28, n. 1, p. 207–216, 2012. ISSN 07475632. Citado 2 vezes nas páginas 29 e 39.

CAUCHY. 1847. Citado na página 61.

CHAE, D.-K.; KIM, S.-W.; LEE, J.-T. Autoencoder-based personalized ranking framework unifying explicit and implicit feedback for accurate top-N recommendation. *Knowledge-Based Systems*, v. 176, p. 110–121, 2019. ISSN 0950-7051. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950705119301510>>. Citado 4 vezes nas páginas 51, 52, 91 e 92.

CHANG, J.-J.; LIN, W.-S.; CHEN, H.-R. How attention level and cognitive style affect learning in a MOOC environment? Based on the perspective of brainwave analysis. *Computers in Human Behavior*, Elsevier, v. 100, n. May 2018, p. 209–217, 2019. ISSN 07475632. Disponível em: <<https://doi.org/10.1016/j.chb.2018.08.016>>. Citado 5 vezes nas páginas 31, 32, 33, 42 e 53.

CHANG, R. I.; HUNG, Y. H.; LIN, C. F. Survey of learning experiences and influence of learning style preferences on user intentions regarding mooc s. *British Journal of Educational Technology*, Wiley Online Library, v. 46, n. 3, p. 528–541, 2015. Citado 2 vezes nas páginas 28 e 33.

CHEN, B.; LEE, C.-Y.; TSAI, I.-C. Ontology-based e-learning system for personalized learning. In: *International Conference on Education, Research and Innovation*. [S.l.: s.n.], 2011. v. 18, p. 38–42. Citado na página 32.

CHEN, L. et al. Personalized itinerary recommendation: Deep and collaborative learning with textual information. *Expert Systems with Applications*, v. 144, p. 113070, 2020. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417419307870>>. Citado 2 vezes nas páginas 34 e 91.

CHEN, S.; PENG, Y. Matrix factorization for recommendation with explicit and implicit feedback. *Knowledge-Based Systems*, v. 158, p. 109–117, 2018. ISSN 0950-7051. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950705118302843>>. Citado 5 vezes nas páginas 51, 52, 63, 91 e 92.

COLINO, F. L. et al. The impact of wellness on neural learning systems. *Neuroscience Letters*, v. 714, p. 134537, 2020. ISSN 0304-3940. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0304394019306408>>. Citado 3 vezes nas páginas 25, 27 e 31.

COUDRAY, C. Bourgault du. Theory and praxis in experiential education: Some insights from gestalt therapy. *Journal of Experiential Education*, SAGE Publications Sage CA: Los Angeles, CA, p. 1053825920904387, 2020. Citado na página 26.

DAWSON, S.; HEATHCOTE, L.; POOLE, G. Harnessing ict potential. *International Journal of Educational Management*, Emerald Group Publishing Limited, 2010. Citado na página 31.

DOCKTERMAN, D. Insights from 200+ years of personalized learning. *npj Science of Learning*, Springer US, v. 3, n. 1, p. 1–6, 2018. ISSN 2056-7936. Disponível em: <<http://dx.doi.org/10.1038/s41539-018-0033-x>>. Citado na página 26.

- DONG, X. et al. A hybrid collaborative filtering model with deep structure for recommender systems. In: *31st AAAI Conference on Artificial Intelligence, AAAI 2017*. [S.l.: s.n.], 2017. p. 1309–1315. Citado 3 vezes nas páginas 51, 52 e 63.
- DRACHSLER, H. et al. Remashed–recommendations for mash-up personal learning environments. In: SPRINGER. *European conference on technology enhanced learning*. [S.l.], 2009. p. 788–793. Citado 2 vezes nas páginas 29 e 51.
- DRACHSLER, H. et al. Panorama of recommender systems to support learning. In: *Recommender systems handbook*. [S.l.]: Springer, 2015. p. 421–451. Citado 2 vezes nas páginas 29 e 34.
- DUTT, A.; ISMAIL, M. A.; HERAWAN, T. A Systematic Review on Educational Data Mining. *IEEE Access*, v. 5, p. 15991–16005, 2017. ISSN 21693536. Citado na página 32.
- EL-BISHOUTY, M. M. et al. Use of felder and silverman learning style model for online course design. *Educational Technology Research and Development*, Springer, v. 67, n. 1, p. 161–177, 2019. Citado na página 47.
- EL-HMOUDOVA, D.; LOUDOVA, I. Implementation of assessment for learning (afl) in blackboard lms and its reflection on tertiary students’ second language performance. In: SPRINGER. *International Symposium on Emerging Technologies for Education*. [S.l.], 2018. p. 23–31. Citado na página 31.
- FELDER, R. M.; SILVERMAN, L. K. et al. Learning and teaching styles in engineering education. *Engineering education*, North Carolina, v. 78, n. 7, p. 674–681, 1988. Citado 11 vezes nas páginas 10, 27, 28, 29, 46, 50, 74, 80, 81, 125 e 146.
- FIORESI, R.; CHAUDHARI, P.; SOATTO, S. A geometric interpretation of stochastic gradient descent using diffusion metrics. *Entropy*, Multidisciplinary Digital Publishing Institute, v. 22, n. 1, p. 101, 2020. Citado 3 vezes nas páginas 60, 61 e 62.
- FRANZONI, A. L. et al. Student learning styles adaptation method based on teaching strategies and electronic media. In: IEEE. *2008 Eighth IEEE International Conference on Advanced Learning Technologies*. [S.l.], 2008. p. 778–782. Citado na página 47.
- FRIEDMAN, L. W.; FRIEDMAN, H. Using social media technologies to enhance online learning. *Journal of Educators Online*, Journal of Educators Online, v. 10, n. 1, p. 1–22, 2013. Citado na página 32.
- GAFFO, F. et al. Edukas environment: Towards an integrated dashboard for education management in smart cities. In: IEEE. *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*. [S.l.], 2017. p. 1–8. Citado 4 vezes nas páginas 35, 41, 42 e 139.
- GAMMERMAN, A.; VOVK, V.; VAPNIK, V. Learning by transduction. In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. (UAI’98), p. 148–155. ISBN 155860555X. Citado na página 64.

- GAO, J.; BARZEL, B.; BARABÁSI, A.-L. Universal resilience patterns in complex networks. *Nature*, Nature Publishing Group, v. 568, n. 7751, p. E5–E5, 2019. Citado na página 34.
- GARRIDO, A.; MORALES, L.; SERINA, I. On the use of case-based planning for e-learning personalization. *Expert Systems with Applications*, v. 60, p. 1–15, 2016. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417416302032>>. Citado na página 54.
- GÖBEL, S.; MEHM, F. Personalized, adaptive digital educational games using narrative game-based learning objects. In: *Serious Games and Virtual Worlds in Education, Professional Development, and Healthcare*. [S.l.]: IGI Global, 2013. p. 74–84. Citado 2 vezes nas páginas 26 e 89.
- GOMEDE, E.; BARROS, R. Miranda de; MENDES, L. de S. Use of deep multi-target prediction to identify learning styles. *Applied Sciences*, v. 10, n. 5, 2020. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/10/5/1756>>. Citado 7 vezes nas páginas 9, 27, 36, 45, 105, 125 e 140.
- GOMEDE, E. et al. Application of computational intelligence to improve education in smart cities. *Sensors (Switzerland)*, v. 18, n. 1, 2018. ISSN 14248220. Citado 11 vezes nas páginas 25, 27, 31, 33, 35, 41, 42, 125, 126, 127 e 139.
- GÓMEZ, S. et al. Context-aware adaptive and personalized mobile learning delivery supported by uolmp. *Journal of King Saud University-Computer and Information Sciences*, Elsevier, v. 26, n. 1, p. 47–61, 2014. Citado na página 26.
- GRAF, S.; KINSHUK; LIU, T.-C. Supporting teachers in identifying students' learning styles in learning management systems: An automatic student modelling approach. *Journal of Educational Technology & Society*, JSTOR, v. 12, n. 4, p. 3–14, 2009. Citado 4 vezes nas páginas 47, 49, 78 e 79.
- HARUNA, K. et al. A collaborative approach for research paper recommender system. *PLoS ONE*, v. 12, n. 10, p. 1–17, 2017. ISSN 19326203. Citado 4 vezes nas páginas 29, 30, 51 e 52.
- HEIDARI, A. A. et al. Ant lion optimizer: theory, literature review, and application in multi-layer perceptron neural networks. In: *Nature-Inspired Optimizers*. [S.l.]: Springer, 2020. p. 23–46. Citado 13 vezes nas páginas 9, 11, 55, 56, 57, 60, 61, 81, 83, 84, 108, 111 e 116.
- HERLOCKER, J. L. et al. An algorithmic framework for performing collaborative filtering. In: ACM NEW YORK, NY, USA. *ACM SIGIR Forum*. [S.l.], 2017. v. 51, n. 2, p. 227–234. Citado na página 40.
- HUANG, C. S. et al. Effects of situated mobile learning approach on learning motivation and performance of efl students. *Journal of Educational Technology & Society*, JSTOR, v. 19, n. 1, p. 263–276, 2016. Citado na página 32.
- HWANG, G.-J.; CHANG, H.-F. A formative assessment-based mobile learning approach to improving the learning attitudes and achievements of students. *Computers & Education*, Elsevier, v. 56, n. 4, p. 1023–1031, 2011. Citado na página 32.

- HWANG, G.-J.; YANG, L.-H.; WANG, S.-Y. A concept map-embedded educational computer game for improving students' learning performance in natural science courses. *Computers & Education*, Elsevier, v. 69, p. 121–130, 2013. Citado na página 32.
- KHAN, S. H.; HAYAT, M.; PORIKLI, F. Regularization of deep neural networks with spectral dropout. *Neural Networks*, Elsevier, v. 110, p. 82–90, 2019. Citado 5 vezes nas páginas 58, 79, 98, 99 e 114.
- KIRAN, R.; KUMAR, P.; BHASKER, B. DNNRec: A novel deep learning based hybrid recommender system. *Expert Systems with Applications*, v. 144, p. 113054, 2020. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417419307717>>. Citado 2 vezes nas páginas 34 e 113.
- KLAŠNJA-MILIĆEVIĆ, A. et al. E-learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, Elsevier, v. 56, n. 3, p. 885–899, 2011. Citado 2 vezes nas páginas 29 e 32.
- KOLB, A. Y.; KOLB, D. A. Learning styles and learning spaces: Enhancing experiential learning in higher education. *Academy of management learning & education*, Academy of Management Briarcliff Manor, NY 10510, v. 4, n. 2, p. 193–212, 2005. Citado 3 vezes nas páginas 27, 45 e 82.
- KRAUSS, C. Smart learning: Time-dependent context-aware learning object recommendations. In: *The Twenty-Ninth International Flairs Conference*. [S.l.: s.n.], 2016. Citado 11 vezes nas páginas 29, 35, 39, 43, 44, 51, 53, 73, 121, 125 e 127.
- KUCHAIEV, O.; GINSBURG, B. Training deep autoencoders for collaborative filtering. *arXiv preprint arXiv:1708.01715*, 2017. Citado 3 vezes nas páginas 58, 116 e 118.
- LAI, J. W.; BOWER, M. How is the use of technology in education evaluated? A systematic review. *Computers and Education*, Elsevier, v. 133, n. January, p. 27–42, 2019. ISSN 03601315. Disponível em: <<https://doi.org/10.1016/j.compedu.2019.01.010>>. Citado na página 31.
- LI, L. X.; RAHMAN, S. S. A. Students' learning style detection using tree augmented naive bayes. *Royal Society open science*, The Royal Society Publishing, v. 5, n. 7, p. 172108, 2018. Citado na página 47.
- LI, X.; SHE, J. Collaborative variational autoencoder for recommender systems. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Part F129685, p. 305–314, 2017. Citado 4 vezes nas páginas 51, 52, 58 e 63.
- LIN, C. F. et al. Data mining for providing a personalized learning path in creativity: An application of decision trees. *Computers & Education*, Elsevier, v. 68, p. 199–210, 2013. Citado na página 26.
- LU, J. et al. Recommender system application developments: A survey. *Decision Support Systems*, Elsevier B.V., v. 74, p. 12–32, 2015. ISSN 01679236. Disponível em: <<http://dx.doi.org/10.1016/j.dss.2015.03.008>>. Citado na página 31.
- MACFADYEN, L. P.; DAWSON, S. Numbers are not enough. why e-learning analytics failed to inform an institutional strategic plan. *Journal of Educational Technology & Society*, JSTOR, v. 15, n. 3, p. 149–163, 2012. Citado na página 31.

- MALONE, T. W.; LAUBACHER, R.; DELLAROCAS, C. Harnessing crowds: Mapping the genome of collective intelligence. MIT Sloan Research Paper, 2009. Citado na página 40.
- MANOUSELIS, N. et al. Recommender systems in technology enhanced learning. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 387–415. Citado 2 vezes nas páginas 29 e 51.
- MARGARYAN, A.; BIANCO, M.; LITTLEJOHN, A. Instructional quality of massive open online courses (moocs). *Computers & Education*, Elsevier, v. 80, p. 77–83, 2015. Citado na página 32.
- MEDIO, C. D. et al. MoodleREC: A recommendation system for creating courses using the moodle e-learning platform. *Computers in Human Behavior*, v. 104, p. 106168, 2020. ISSN 0747-5632. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0747563219303802>>. Citado 5 vezes nas páginas 52, 53, 121, 125 e 127.
- MILLER, G. A. et al. Individual differences in imagery and the psychophysiology of emotion. *Cognition and Emotion*, Taylor & Francis, v. 1, n. 4, p. 367–390, 1987. Citado na página 33.
- MONGIA, A. et al. Deep latent factor model for collaborative filtering. *Signal Processing*, v. 169, p. 107366, 2020. ISSN 0165-1684. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0165168419304190>>. Citado na página 34.
- MOUBAYED, A. et al. e-Learning: Challenges and Research Opportunities Using Machine Learning & Data Analytics. *IEEE Access*, IEEE, v. 6, p. 39117–39138, 2018. ISSN 21693536. Citado 5 vezes nas páginas 30, 31, 32, 33 e 34.
- NAFEA, S. M.; SIEWE, F.; HE, Y. On Recommendation of Learning Objects using Felder-Silverman Learning Style Model. *IEEE Access*, v. 4, p. 1–1, 2019. ISSN 21693536. Citado 7 vezes nas páginas 27, 29, 30, 52, 53, 121 e 125.
- NASSAR, N.; JAFAR, A.; RAHHAL, Y. A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowledge-Based Systems*, v. 187, p. 104811, 2020. ISSN 0950-7051. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950705119302862>>. Citado na página 34.
- NORMADHI, N. B. A. et al. Identification of personal traits in adaptive learning environment: Systematic literature review. *Computers & Education*, Elsevier, v. 130, p. 168–190, 2019. Citado 19 vezes nas páginas 27, 29, 30, 31, 33, 34, 39, 40, 41, 42, 43, 44, 46, 48, 54, 75, 82, 111 e 124.
- ORDWAY-WEST, E.; PARVEEN, P.; HENSLEE, A. Autoencoder evaluation and hyper-parameter tuning in an unsupervised setting. In: IEEE. *2018 IEEE International Congress on Big Data (BigData Congress)*. [S.l.], 2018. p. 205–209. Citado 3 vezes nas páginas 15, 58 e 99.
- PARAMYTHIS, A.; LOIDL-REISINGER, S. Adaptive learning environments and e-learning standards. In: *Second european conference on e-learning*. [S.l.: s.n.], 2003. v. 1, n. 2003, p. 369–379. Citado na página 30.

- PASHLER, H. et al. Learning styles: Concepts and evidence. *Psychological science in the public interest*, SAGE Publications Sage CA: Los Angeles, CA, v. 9, n. 3, p. 105–119, 2008. Citado 3 vezes nas páginas 28, 45 e 50.
- PEÑA-AYALA, A. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Systems with Applications*, v. 41, n. 4 PART 1, p. 1432–1462, 2014. ISSN 09574174. Citado na página 31.
- PROTASIEWICZ, J. et al. A recommender system of reviewers and experts in reviewing problems. *Knowledge-Based Systems*, v. 106, p. 164–178, 2016. ISSN 0950-7051. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950705116301381>>. Citado 4 vezes nas páginas 52, 53, 121 e 125.
- PSOTKA, J.; MASSEY, L. D.; MUTTER, S. A. *Intelligent tutoring systems: Lessons learned*. [S.l.]: Psychology Press, 1988. Citado na página 30.
- RASCHKA, S. Model evaluation, model selection, and algorithm selection in machine learning. *ArXiv*, abs/1811.12808, 2018. Citado 10 vezes nas páginas 9, 10, 48, 50, 65, 66, 67, 68, 69 e 70.
- RASHEED, F.; WAHID, A. Learning style recognition: A neural network approach. In: SPRINGER. *First International Conference on Artificial Intelligence and Cognitive Computing*. [S.l.], 2019. p. 301–312. Citado na página 47.
- RODRIGUES, M. W.; ISOTANI, S.; ZÁRATE, L. E. Educational Data Mining: A review of evaluation process in the e-learning. *Telematics and Informatics*, Elsevier, v. 35, n. 6, p. 1701–1717, 2018. ISSN 07365853. Disponível em: <<https://doi.org/10.1016/j.tele.2018.04.015>>. Citado na página 31.
- ROMERO, C.; VENTURA, S. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, v. 33, n. 1, p. 135–146, 2007. ISSN 09574174. Citado 2 vezes nas páginas 31 e 34.
- ROMERO, C.; VENTURA, S. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, v. 40, n. 6, p. 601–618, 2010. ISSN 10946977. Citado 5 vezes nas páginas 27, 31, 32, 34 e 39.
- SADLER-SMITH, E. The relationship between learning style and cognitive style. *Personality and Individual Differences*, v. 30, p. 609–616, 03 2001. Citado na página 27.
- SARYAR, S. et al. Mobile learning recommender system based on learning styles. In: *Soft Computing and Signal Processing*. [S.l.]: Springer, 2019. p. 299–312. Citado na página 47.
- SATHYA, R.; ABRAHAM, A. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, Citeseer, v. 2, n. 2, p. 34–38, 2013. Citado 2 vezes nas páginas 63 e 64.
- SCHUNK, D. H.; MEECE, J. R.; PINTRICH, P. R. *Motivation in education: Theory, research, and applications*. [S.l.]: Pearson Higher Ed, 2012. Citado 2 vezes nas páginas 25 e 26.

- SEDHAIN, S. et al. AutoRec: Autoencoders meet collaborative filtering. *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web*, p. 111–112, 2015. Citado 3 vezes nas páginas 51, 52 e 58.
- SHEEBA, T.; KRISHNAN, R. Automatic Detection of Students Learning Style in Learning Management System. Springer International Publishing, p. 333–338, 2019. Disponível em: <<http://link.springer.com/10.1007/978-3-030-01659-3>>. Citado 3 vezes nas páginas 31, 49 e 78.
- SHIH, J.-L.; CHUANG, C.-W.; HWANG, G.-J. An inquiry-based mobile learning approach to enhancing social science learning effectiveness. *Journal of Educational Technology & Society*, JSTOR, v. 13, n. 4, p. 50–62, 2010. Citado na página 32.
- SILVA, J. F. G. da; JUNIOR, N. N. de M.; CALOBA, L. P. Effects of data sparsity on recommender systems based on collaborative filtering. In: IEEE. *2018 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2018. p. 1–8. Citado 2 vezes nas páginas 63 e 93.
- SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, v. 15, p. 1929–1958, 2014. Disponível em: <<http://jmlr.org/papers/v15/srivastava14a.html>>. Citado 2 vezes nas páginas 79 e 114.
- STRUB, F.; GAUDEL, R.; MARY, J. Hybrid recommender system based on autoencoders. *ACM International Conference Proceeding Series*, v. 15-September-2016, p. 11–16, 2016. Citado 3 vezes nas páginas 51, 52 e 58.
- TANG, T.; MCCALLA, G. Smart recommendation for an evolving e-learning system: Architecture and experiment. *International Journal on E-learning*, Association for the Advancement of Computing in Education (AACE), v. 4, n. 1, p. 105–129, 2005. Citado 3 vezes nas páginas 29, 42 e 51.
- TOVEN-LINDSEY, B.; RHOADS, R. A.; LOZANO, J. B. Virtually unlimited classrooms: Pedagogical practices in massive open online courses. *The internet and higher education*, Elsevier, v. 24, p. 1–12, 2015. Citado na página 32.
- TRAUSAN-MATU, S.; STAHL, G.; SARMIENTO, J. Polyphonic support for collaborative learning. In: SPRINGER. *International Conference on Collaboration and Technology*. [S.l.], 2006. p. 132–139. Citado na página 40.
- TRUONG, H. M. Integrating learning styles and adaptive e-learning system: Current developments, problems and opportunities. *Computers in human behavior*, Elsevier, v. 55, p. 1185–1193, 2016. Citado 5 vezes nas páginas 44, 46, 47, 48 e 110.
- VESIN, B. et al. Applying recommender systems and adaptive hypermedia for e-learning personalization. *Computing and informatics*, v. 32, n. 3, p. 629–659, 2013. Citado 3 vezes nas páginas 29, 40 e 41.
- VOLKWYN, T. S. et al. Transduction and science learning: Multimodality in the physics laboratory. *Designs for Learning*, v. 11, n. 1, p. 16–29, 2019. Citado na página 64.

- WANG, H.; WANG, N.; YEUNG, D.-Y. Collaborative Deep Learning for Recommender Systems. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2015. (KDD '15), p. 1235–1244. ISBN 9781450336642. Disponível em: <<https://doi.org/10.1145/2783258.2783273>>. Citado 13 vezes nas páginas 34, 42, 51, 52, 58, 88, 89, 94, 96, 97, 116, 118 e 121.
- WANG, S. L.; WU, C. Y. Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. *Expert Systems with Applications*, Elsevier Ltd, v. 38, n. 9, p. 10831–10838, 2011. ISSN 09574174. Disponível em: <<http://dx.doi.org/10.1016/j.eswa.2011.02.083>>. Citado 5 vezes nas páginas 34, 42, 52, 53 e 125.
- WILLINGHAM, D. T. The Digital Expansion of the Mind Gone Wrong in Education. *Journal of Applied Research in Memory and Cognition*, v. 8, n. 1, p. 20–24, 2019. ISSN 2211-3681. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2211368118302778>>. Citado 10 vezes nas páginas 25, 27, 28, 29, 31, 41, 42, 45, 74 e 124.
- WILLINGHAM, D. T.; HUGHES, E. M.; DOBOLYI, D. G. The Scientific Status of Learning Styles Theories. *Teaching of Psychology*, v. 42, n. 3, p. 266–271, 2015. ISSN 15322802. Citado 9 vezes nas páginas 25, 26, 27, 28, 41, 42, 45, 52 e 74.
- WU, W.-H. et al. Review of trends from mobile learning studies: A meta-analysis. *Computers & Education*, Elsevier, v. 59, n. 2, p. 817–827, 2012. Citado na página 32.
- WU, Y. et al. Collaborative denoising auto-encoders for top-n recommender systems. In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. [S.l.: s.n.], 2016. p. 153–162. Citado 2 vezes nas páginas 58 e 113.
- XU, Y.; GOODACRE, R. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, v. 2, n. 3, p. 249–262, Jul 2018. ISSN 2509-4696. Disponível em: <<https://doi.org/10.1007/s41664-018-0068-2>>. Citado 3 vezes nas páginas 66, 67 e 69.
- XUE, H. J. et al. Deep Matrix Factorization Models for Recommender Systems. *IJCAI International Joint Conference on Artificial Intelligence*, p. 3203–3209, 2017. ISSN 10450823. Citado 3 vezes nas páginas 51, 52 e 92.
- YU, H. et al. Towards AI-powered personalization in MOOC learning. *npj Science of Learning*, Springer US, v. 2, n. 1, p. 1–5, 2017. ISSN 2056-7936. Disponível em: <<http://dx.doi.org/10.1038/s41539-017-0016-3>>. Citado na página 31.
- YU, K. et al. Effects of the application of multimedia to library use education on learning motivation and learning satisfaction. *Eurasia Journal of Mathematics, Science and Technology Education*, Modestum LTD, v. 14, n. 7, p. 2987–2994, 2018. Citado na página 31.
- ZHANG, S. et al. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 52, n. 1, 2019. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/3285029>>. Citado na página 51.

ZHAO, Q. et al. Explicit or Implicit Feedback? Engagement or Satisfaction? A Field Experiment on Machine-Learning-Based Recommender Systems. *Sac*, n. 10, 2018. Disponível em: <<https://doi.org/10.1145/3167132.3167275>>. Citado 3 vezes nas páginas 51, 52 e 91.

ZUO, Y. et al. Tag-aware recommender systems based on deep neural networks. *Neurocomputing*, v. 204, p. 51–60, 2016. ISSN 0925-2312. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925231216301151>>. Citado 2 vezes nas páginas 34 e 63.

Apêndices

APÊNDICE A – Trabalhos Publicados pelo Autor

Edukas Environment: Towards an Integrated Dashboard for Education Management in Smart Cities. Education is considered one of the basic needs of every citizen and plays a key role in country development. Throughout history this area has faced a number of challenges and paradigm shifts. Nowadays, the presence of pervasive computing might help the teaching-learning process using huge datasets through user interaction with these new technologies. This research describes the Edukas Environment, a conceptual model that uses educational governance, computational intelligence and strategic planning to assist school and smart city managers transform the teaching-learning process, benefiting the whole educational chain of events (GAFFO et al., 2017).

Disponível em: <https://ieeexplore.ieee.org/document/8397535>.

Application of Computational Intelligence to Improve Education in Smart Cities. According to UNESCO, education is a fundamental human right and every nation's citizens should be granted universal access with equal quality to it. Because this goal is yet to be achieved in most countries, in particular in the developing and underdeveloped countries, it is extremely important to find more effective ways to improve education. This paper presents a model based on the application of computational intelligence (data mining and data science) that leads to the development of the student's knowledge profile and that can help educators in their decision making for best orienting their students. This model also tries to establish key performance indicators to monitor objectives' achievement within individual strategic planning assembled for each student. The model uses random forest for classification and prediction, graph description for data structure visualization and recommendation systems to present relevant information to stakeholders. The results presented were built based on the real dataset obtained from a Brazilian private k-9 (elementary school). The obtained results include correlations among key data, a model to predict student performance and recommendations that were generated for the stakeholders (GOMEDE et al., 2018).

Disponível em: <https://www.mdpi.com/1424-8220/18/1/267>.

Use of Deep Multi Target Prediction to Identify Learning Styles. It is possible to classify students according to the manner they recognize, process and store information. This classification should be considered when developing adaptive e-

learning systems. It also creates a comprehension on the different styles students demonstrate while in the process of learning, which can help adaptive e-learning systems offer advice and instructions to students, teachers, administrators, and parents in order to optimize students learning processes. Moreover, e-learning systems using computational and statistical algorithms to analyze students learning may offer the opportunity to complement traditional learning evaluation methods with new ones based on analytical intelligence. In this work, we propose a method based on deep multi target prediction algorithm using Felder-Silverman learning styles model to improve students' learning evaluation using feature selection, learning styles models, and multiple target classification. As a result, we present a set of features and a model based on artificial neural network to investigate the possibility of improving the accuracy of automatic learning styles identification. The obtained results show that learning styles allows adaptive e-learning systems improve the learning process of students (GOMEDE; BARROS; MENDES, 2020).

Disponível em: <https://www.mdpi.com/2076-3417/10/5/1756>.

APÊNDICE B – Trabalhos Aguardando a Publicação

Deep Auto Encoder to Adaptation in E-learning Recommender System. The adaptive learning, supported by information & communication technology, is an important research area of educational systems to improve the outcomes of students. Thus, to investigate what and how to adapt to constitute a foundation of adaptive e-learning systems (AES). In this paper, we compared three classes of DEA and the popularity model to address the problem to explain and predict the student preference in the AES; Collaborative Denoising Auto Encoders (CDAE), Deep Auto Encoders for Collaborative Filtering (DAE-CF), Deep Auto Encoders for Collaborative Filtering with Content Information (DAE-CI). The results point out that the DAE-CF is more powerful to provide significant adaptation. Furthermore, we present a concept of student signature to represent the most granular class of adaptation. Therefore, this model may be using in e-learning systems to provide adaptation and help to improve the outcomes of students. (2020)

APÊNDICE C – Taxonomia de Sistemas de Recomendação

Os sistemas de recomendação personalizados, baseados em modelos de fatores latentes, são amplamente utilizados para aumentar as vendas no comércio eletrônico. Estes sistemas usam o comportamento passado dos usuários para recomendar novos itens que provavelmente lhes interessam.

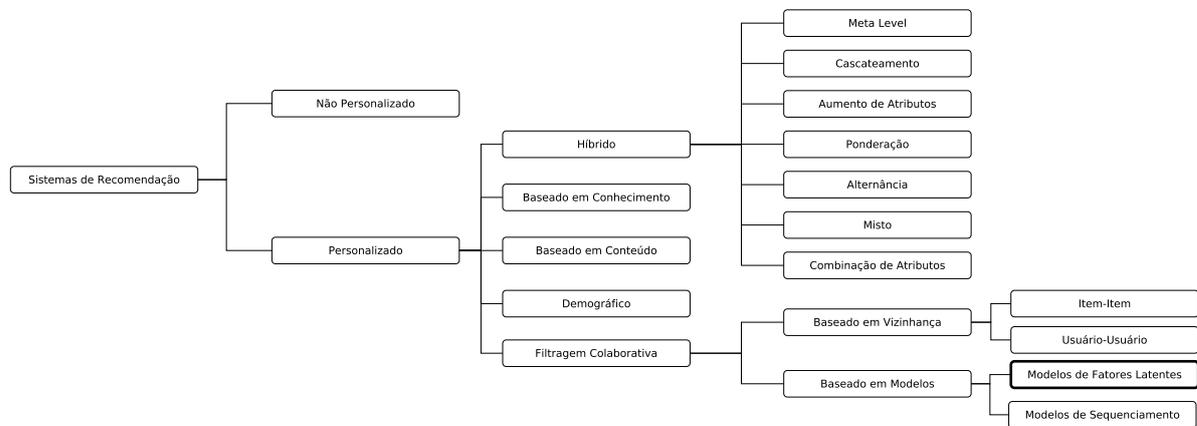


Figura C.1 – Taxonomia para as técnicas empregadas em sistemas de recomendação. Dentre cada um dos grupos pode se utilizar algoritmos de redes neurais entre outros.

APÊNDICE D – Visão Geral do Experimento de Classificação

A visão geral do experimento de classificação pode ser dada como segue. O comportamento do aluno é capturado, via interação com os objetos de aprendizagem contidos no MOOC dentro do LMS, e armazenado em uma matriz. Posteriormente, os dados são rotulados de acordo com o questionário de estilos de aprendizagem. Depois disso, cada característica relacionada com os objetos de aprendizagem, por exemplo, *content_visit*, é associada com um neurônio na camada de entrada. A rede é, então, treinada para gerar a saída múltipla que mais se aproxima com os estilos das observações de comportamento de alunos rotulados. Esta visão geral pode ser vista na Figura D.1.

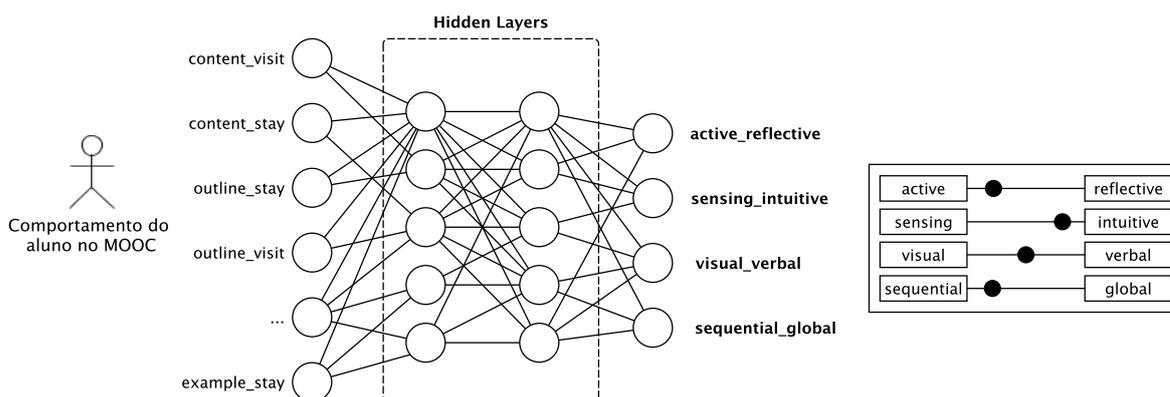


Figura D.1 – A visão geral do experimento de classificação. As entradas da rede representam o comportamento do aluno ao interagir com os objetos de aprendizagem. A saída representa os estilos de aprendizagem do FSLSM.

APÊNDICE E – Visão Geral do Experimento de Recomendação

A visão geral do experimento de recomendação pode ser dada como segue. O comportamento do aluno é capturado, via interação com os objetos de aprendizagem contidos no MOOC dentro do LMS, e armazenado em uma matriz, geralmente, esparsa. Posteriormente, os dados são normalizados e fornecidos como entradas da rede neural. A rede é então treinada para reconstruir as interações faltantes. Com isso, a assinatura do aluno é construída e pode prever o comportamento deste. Esta visão geral pode ser vista na Figura E.1.

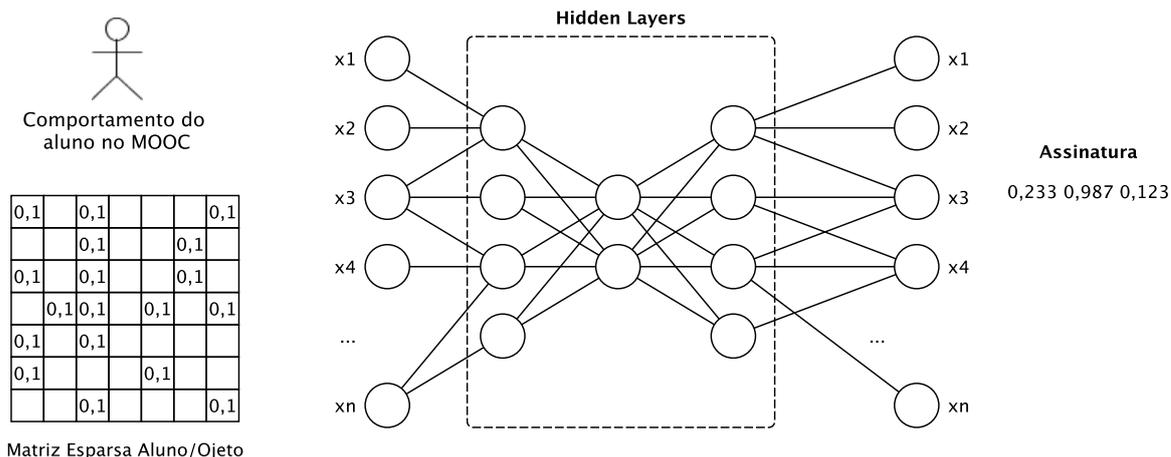


Figura E.1 – A visão geral do experimento de recomendação. A entrada da rede representa o comportamento do aluno ao interagir com os objetos de aprendizagem. A saída esta associada com a previsão do comportamento do mesmo.

Anexos

ANEXO A – Questionário para Identificação de Estilos de Aprendizagem

A Figura D.1 mostra o resultado da detecção de estilos de aprendizagem, baseado no modelo FLSM (FELDER; SILVERMAN et al., 1988). Pode-se notar as preferências do aluno nas quatro dimensões do modelo. Isso permite uma melhor adequação dos formatos dos materiais a serem entregues, agrupamento de alunos por similaridade e identificação do perfil do aluno. O questionário a seguir foi extraído da pagina oficial de Felder-Silverman ¹ e utilizado para rotular os dados do experimento de classificação.

Questionnaire Results for Everton Gomedé:

> Reflective: 9 > Intuitive: 9 > Visual: 9 > Global: 9

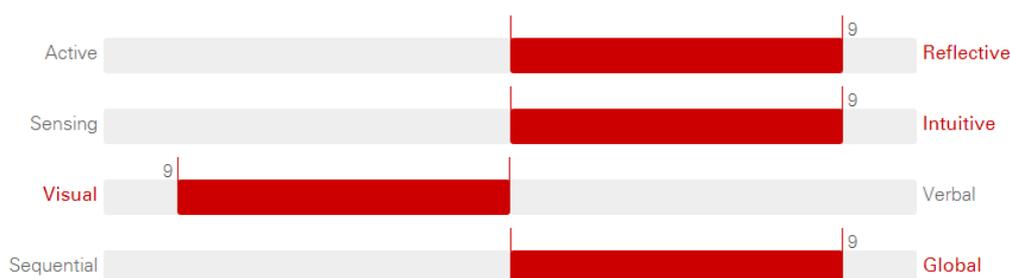


Figura A.1 – Resultado da aplicação do questionário do modelo de estilos de aprendizagem de Felder-Silverman. Os resultados para cada dimensão indicam a tendência na preferência do aluno.

¹ <https://www.webtools.ncsu.edu/learningstyles/>

Tabela A.1 – Questionário de avaliação dos estilos de aprendizagem – 1/4

I understand something better after I
try it out think it through
I would rather be considered
realistic innovative
When I think about what I did yesterday, I am most likely to get
a picture words
I tend to
understand details of a subject but may be fuzzy about its overall structure understand the overall structure but may be fuzzy about details
When I am learning something new, it helps me to
talk about it think about it
If I were a teacher, I would rather teach a course
that deals with facts and real life situations that deals with ideas and theories
I prefer to get new information in
pictures, diagrams, graphs, or maps written directions or verbal information
Once I understand
all the parts, I understand the whole thing the whole thing, I see how the parts fit
In a study group working on difficult material, I am more likely to
jump in and contribute ideas sit back and listen
I find it easier
to learn facts to learn concepts
In a book with lots of pictures and charts, I am likely to
look over the pictures and charts carefully focus on the written text

Tabela A.2 – Questionário de avaliação dos estilos de aprendizagem – 2/4

When I solve math problems

I usually work my way to the solutions one step at a time

I see the solutions but then have to struggle to figure out the steps to get to them

In classes I have taken

I have usually gotten to know many of the students

I have rarely gotten to know many of the students

In reading nonfiction, I prefer

something that teaches me new facts or tells me how to do something

something that gives me new ideas to think about

I like teachers

who put a lot of diagrams on the board

who spend a lot of time explaining

When I'm analyzing a story or a novel

I think of the incidents and try to put them together to figure out the themes

I know the themes are when I finish then I go back and find that demonstrate them.

When I start a homework problem, I am more likely to

start working on the solution immediately

try to fully understand the problem first

I prefer the idea of

certainty

theory

I remember best

what I see

what I hear

It is more important to me that an instructor

lay out the material in clear sequential steps

give me an overall picture and relate the material to other subjects

I prefer to study

in a study group

alone

I am more likely to be considered

careful about the details of my work

creative about how to do my work

Tabela A.3 – Questionário de avaliação dos estilos de aprendizagem – 3/4

When I get directions to a new place, I prefer
a map written directions
I learn
at a fairly regular pace. If I study hard, I'll "get it." in fits and starts. I'll be totally confused and then suddenly it all "clicks."
I would rather first
try things out think about how I'm going to do it
When I am reading for enjoyment, I like writers to
clearly say what they mean say things in creative, interesting ways
When I see a diagram or sketch in class, I am most likely to remember
the picture what the instructor said about it
When considering a body of information, I am more likely to
focus on details and miss the big picture try to understand the big picture before getting into the details.
I more easily remember
something I have done something I have thought a lot about
When I have to perform a task, I prefer to
master one way of doing it come up with new ways of doing it
When someone is showing me data, I prefer
charts or graphs text summarizing the results
When writing a paper, I am more likely to
work on (think about or write) the beginning of the paper and progress forward work on (think about or write) different parts of the paper and then order them
When I have to work on a group project, I first want to
have "group brainstorming" where everyone contributes ideas brainstorm individually and then come together as a group to compare ideas

Tabela A.4 – Questionário de avaliação dos estilos de aprendizagem – 4/4

I consider it higher praise to call someone

sensible
imaginative

When I meet people at a party, I am more likely to remember

what they looked like
what they said about themselves

When I am learning a new subject, I prefer to

stay focused on that subject, learning as much about it as I can
try to make connections between that subject and related subjects

I am more likely to be considered

outgoing
reserved

I prefer courses that emphasize

concrete material (facts, data)
abstract material (concepts, theories)

For entertainment, I would rather

watch television
read a book

Some teachers start their lectures with an outline. Such outlines are

somewhat helpful to me
very helpful to me

The idea of doing homework in groups, with one grade for the entire group,

appeals to me
does not appeal to me

When I am doing long calculations,

I tend to repeat all my steps and check my work carefully
I find checking my work tiresome and have to force myself to do it

I tend to picture places I have been

easily and fairly accurately
with difficulty and without much detail

When solving problems in a group, I would be more likely to

think of the steps in the solutions process
think of possible consequences or applications of the solution in a wide range of areas