

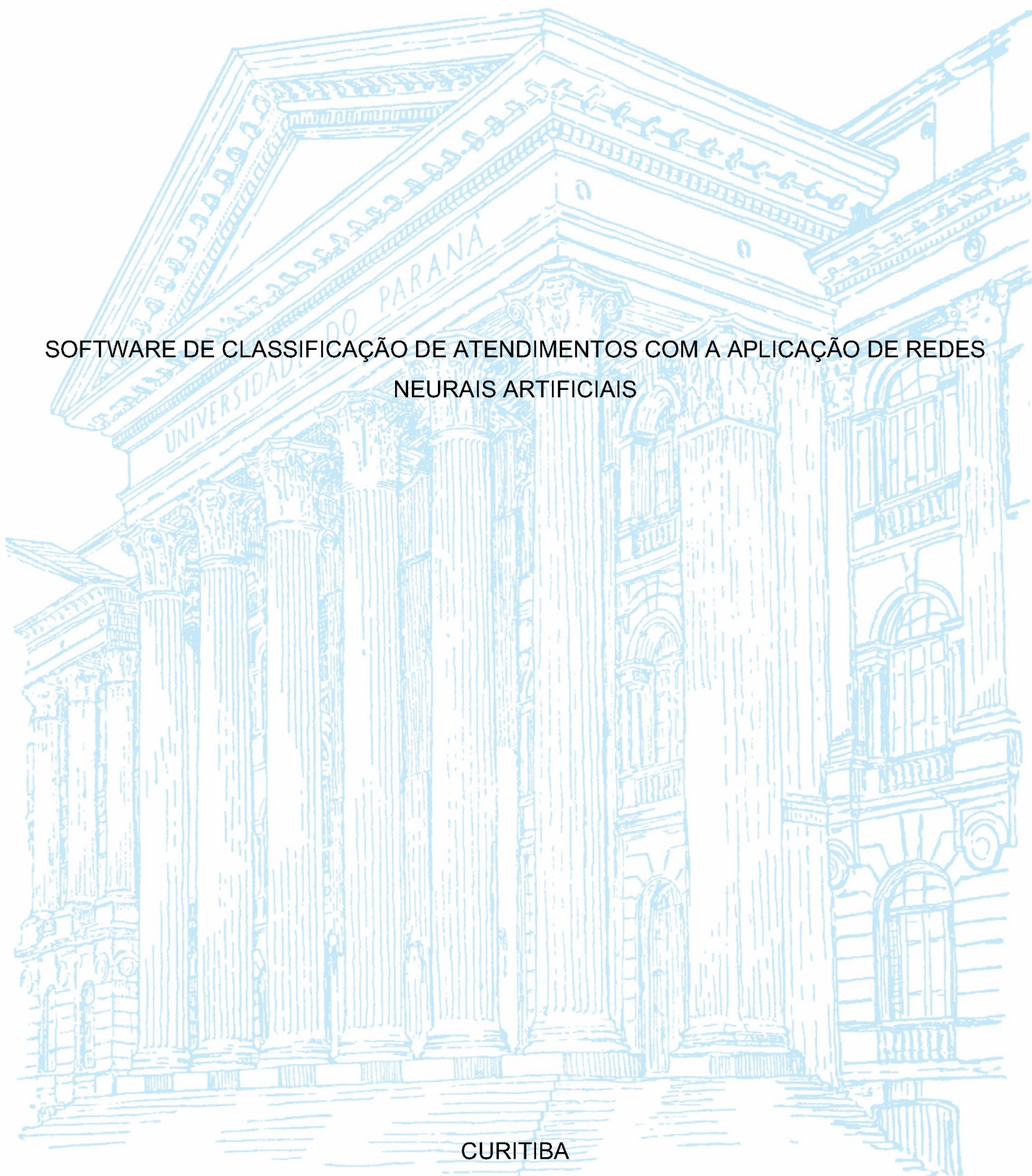
UNIVERSIDADE FEDERAL DO PARANÁ

RODRIGO TELEGINSKI VIDAL

SOFTWARE DE CLASSIFICAÇÃO DE ATENDIMENTOS COM A APLICAÇÃO DE REDES
NEURAS ARTIFICIAIS

CURITIBA

2022



RODRIGO TELEGINSKI VIDAL

SOFTWARE DE CLASSIFICAÇÃO DE ATENDIMENTOS COM A APLICAÇÃO DE REDES
NEURAS ARTIFICIAIS

Trabalho de Conclusão de Curso apresentada ao curso de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada.

Orientador: Prof. Dr. Jaime Wojciechowski

CURITIBA

2022

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **RODRIGO TELEGINSKI VIDAL** intitulada : **Software de Classificação de Atendimentos com a Aplicação de Redes Neurais Artificiais**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO ___ no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 25 de Novembro de 2022.



JAIME WOJCIECHOWSKI
Presidente da Banca Examinadora



RAZER ANTHOM NIZER ROJAS MONTAÑO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Software de Classificação de Atendimentos com a Aplicação de Redes Neurais Artificiais

Rodrigo Teleginski Vidal
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
rodrigoteleginski8@gmail.com

Jaime Wojciechowski
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
jaimewo@ufpr.br

Resumo—Este artigo descreve o desenvolvimento de um software como o objetivo de classificar os atendimentos realizados nas escolas que oferecem programas de educação tecnológica por meio da utilização de técnicas de inteligência artificial. Utilizou-se como base os relatórios realizados pelos orientadores educacionais em um sistema próprio da empresa em questão. Após a extração, agrupamento e classificação, aplicou-se um algoritmo de Rede Neural Artificial. Com isso, foi possível obter um modelo de predição baseado na métrica de precisão para utilização na construção do software. A classificação tem como objetivo fornecer aos coordenadores métricas para melhoria das assessorias. Como resultado, obteve-se um modelo preciso o suficiente em relação a base de dados disponível e a documentação do funcionamento do software.

Palavras-chave—Software de IA, educação tecnológica, qualidade de atendimento, RNA

Abstract—This paper describes the development of a software program with the objective of classifying the attendance performed in schools that offer technological education programs using artificial intelligence techniques. It uses as a base the reports made by the educational advisors in a proprietary system of the company in question. After extraction, clustering and classification, an Artificial Neural Network algorithm was applied. With this, it was possible to obtain a prediction model based on the metric of precision for use in the construction of the software. The classification has the objective of providing coordinators with metrics for the improvement of the advisories. As a result, a model was obtained that is accurate enough in relation to the available database and the documentation of how the software works.

Index Terms—AI software, technology education, quality of attendance, ANN

I. DESENVOLVIMENTO

A tecnologia vem se tornando uma ferramenta cada vez mais importante no processo de aprendizagem dos estudantes do ensino fundamental [1]. Esse movimento pode ser chamado de cultura digital e foi homologada pela Base Nacional Comum Curricular (BNCC) implantada pelo Ministério da Educação em 2017, tornando assim, fundamental trabalhar habilidades e competências para além do conteúdo curricular [2]. Como por exemplo, as disciplinas de STEM (Ciências, Tecnologia, Engenharia e Matemática) que por meio da tecnologia, promovem aos alunos aprendizagem prática e por experimentação [1].

Neste contexto, a ZOOM Education¹, uma empresa brasileira que desenvolve há 24 anos projetos de educação tecnológica para escolas de ensino fundamental, vem

enfrentando dificuldades em como medir a qualidade da execução dos programas ofertados aos seus clientes.

Com isso, este relatório técnico explora o desenvolvimento de um software de inteligência artificial para classificar o desempenho do projeto nas escolas parceiras. Utilizou como base os relatórios de atendimento preenchidos pelos orientadores educacionais, que por sua vez, foram classificados por meio de técnicas de agrupamento e serviram como base para criação de um modelo de predição com Redes Neurais Artificiais. O software servirá como um painel de acompanhamento apresentando o desempenho do projeto em cada instituição com uma contagem de atendimentos ótimos, regulares e ruins. Também conta com uma espécie de termômetro geral baseado na média da classificação de cada atendimento. Com isso, os coordenadores da empresa, podem obter métricas de desempenho.

A. Descrição dos Dados

Para a construção do modelo de predição utilizado neste software, foi utilizada uma base de registros de atendimentos do aplicativo próprio da ZOOM Education. É nele que os orientadores educacionais registram as assessorias e os coordenadores realizam o acompanhamento do projeto nas escolas. Esse software funciona como um sistema de registros em formato de formulário, com dados da assessoria e vários campos de *checkbox* com atividades ou ações que podem ser realizados durante o atendimento.

A base de dados de registros que foi utilizada, agrega os atendimentos de todas os parceiros de janeiro de 2021 até julho de 2022 com 708 registros e 320 variáveis. Esses registros foram agrupados e classificados, sendo 517 como ótimo, 114 como regular e 77 como ruim. Para cada registro, apresentam-se os atributos descritos na Tabela I.

B. Métodos

Para a construção do software em questão, foram aplicados métodos que são descritos nesta seção. Esse procedimento foi realizado em 8 etapas: 1) extração da base de dados de atendimentos; 2) pré-processamento dos dados; 3) aplicação de técnicas de agrupamento; 4) classificação dos dados para treinamento; 5) preparação para treinamento; 6) treinamento do modelo e escolha dos parâmetros; 7) predição; 8) construção do software. Após isso, tem-se um aplicativo que processa os atendimentos das escolas, classifica-os e gera uma visualização para o usuário sobre a qualidade do projeto. As etapas citadas anteriormente estão descritas a seguir:

¹ <https://zoom.education/>

1) *Extração dos atendimentos*: Como início do processo, foram extraídos todos os registros de atendimentos presentes na base de dados do software próprio da empresa, do ano de 2021 ao final do mês de julho de 2022, data em que esse trabalho foi iniciado.

2) *Pré-processamento*: Nesta etapa, são organizados os dados que serão utilizados pelo algoritmo de agrupamento.

Foram removidas todas as variáveis que se referenciavam a campos de texto, haja vista que continham campos como data, protocolo, nome do orientador e um campo de descrição do atendimento. Campos esses que não demonstram ter relação com o desempenho do atendimento não trazendo necessidade de inclusão. Mantendo-se apenas registros que contiam dados numéricos.

Além disso, os dados ainda estavam salvos no banco de dados em um formato diferente do que seria utilizado para a criação de um *Data Frame* [3] (estrutura em formato de tabela utilizada na linguagem R). Os diversos campos de *checkbox* de uma mesma seção, como por exemplo os de dificuldades de atendimento da educação infantil, estavam salvos em uma mesmo campo no banco de dados. O resultado da marcação de um *checkbox* pode ser interpretado como 0, para não marcado, e 1 para marcado. Com isso, uma longa *string* era armazenada com zeros e uns para aquela seção. Então foi necessário que colunas, que continham dados aninhados, fossem fragmentadas. Ou seja, cada 0 ou 1 ficasse individualmente representado por uma coluna, onde inicialmente havia 24 atributos do sistema utilizado internamente, resultou-se em 320 variáveis individuais.

3) *Aplicação do agrupamento*: Com base nos atendimentos extraídos, aplicou-se o algoritmo de agrupamento *k-means* [4], a fim de gerar grupos para classificação. Foi definido, em conversa com os coordenadores, a quantidade ideal de 3 grupos, para atendimentos considerados ótimos, regulares ou ruins.

Esse algoritmo de agrupamento funciona calculando a distância entre os registros dispostos em um plano cartesiano. Ele define centroides para os grupos que vão sendo atualizadas a medida que o algoritmo vai sendo executado. O agrupamento acontece pela distância das ocorrências em relação aos centroides das classes. [4]

Após rodar esse algoritmo, os grupos gerados eram similares e foram nomeados inicialmente como 1, 2 e 3. Contendo respectivamente 76, 518 e 114 registros.

4) *Classificação dos dados para agrupamento*: Após o agrupamento, uma análise sobre as classes geradas foi realizada para definir qual se encaixa nos padrões de ótimo, regular ou ruim. Essa análise e definição foi realizada pelos próprios coordenadores pedagógicos. Com isso, um novo atributo foi inserido na base de dados contendo a classificação do registro com base em seu grupo definido anteriormente. Resultando em uma base de 321 variáveis representado pela Tabela I.

TABELA I
ATRIBUTOS DA BASE DE DADOS

Atributo	Descrição
<i>apoio_gestao</i>	Grau de apoio da gestão da escola para com o projeto (1 a 5)
<i>apoio_escola</i>	Grau de apoio dos professores da escola para com o projeto (1 a 5)
<i>acao_1</i>	Se foi realizada formação pedagógica
<i>acao_2</i>	Se foi realizado diagnóstico de materiais da escola
<i>acao_3</i>	Se foi realizada assessoria pedagógica
<i>acao_4</i>	Se foi realizada reunião com a equipe gestora
<i>acao_5</i>	Se foi realizado a entrega de algum material
<i>acao_6</i>	Se foi realizado torneio
<i>acao_7</i>	Se foi realizado reunião de pais
<i>acao_8</i>	Se foi realizado acompanhamento da aula
<i>acao_9</i>	Se foi realizado evento
<i>acao_10</i>	Se foi realizado conferência e organização de materiais
<i>gestao_1</i>	Se foi realizado formação com a equipe gestora
<i>gestao_2</i>	Se foi abordado os princípios do programa e BNCC
<i>gestao_3</i>	Se foi realizado devolutiva do Programa/assessoria
<i>gestao_4</i>	Se foi realizado cronograma de desenvolvimento das aulas
<i>gestao_5</i>	Se foi realizado o planejamento das assessorias
<i>gestao_6</i>	Se foi abordado as plataformas digitais
<i>gestao_7</i>	Se foi abordado as trilhas digitais
<i>gestao_8</i>	Se foi abordado a feira de ciências
<i>gestao_9</i>	Se foi abordado a mostra cultural
<i>gestao_10</i>	Se foi abordado o torneio interno
<i>gestao_11</i>	Se foi abordado o torneio externo
<i>gestao_12</i>	Se foi realizado fidelização/ampliação de contrato
<i>diffZei_1 até diffZei_21</i>	Representam 21 pontos de dificuldades que podem ser encontrados na assessoria da Educação Infantil
<i>assZei_1 até assZei_19</i>	Representam 19 pontos de assuntos que podem ser abordados na assessoria da Educação Infantil
<i>diff1e2_1 até diff1e2_25</i>	Representam 25 pontos de dificuldades que podem ser encontrados na assessoria do 1º e 2º Ano
<i>ass1e2_1 até ass1e2_26</i>	Representam 26 pontos de assuntos que podem ser abordados na assessoria do 1º e 2º Ano
<i>diff3a5_1 até diff3a5_26</i>	Representam 26 pontos de dificuldades que podem ser encontrados na assessoria do 3º a 5º Ano

<i>ass3a5_1 até ass3a5_25</i>	Representam 25 pontos de assuntos que podem ser abordados na assessoria do 3º a 5º Ano
<i>diff6e7_1 até ass6e7_27</i>	Representam 27 pontos de dificuldades que podem ser encontrados na assessoria do 6º e 7º Ano
<i>ass6e7_1 até ass6e7_27</i>	Representam 27 pontos de assuntos que podem ser abordados na assessoria do 6º e 7º Ano
<i>diff8e9_1 até diff8e9_27</i>	Representam 27 pontos de dificuldades que podem ser encontrados na assessoria do 8º e 9º Ano
<i>ass8e9_1 até ass8e9_28</i>	Representam 28 pontos de assuntos que podem ser abordados na assessoria do 8º e 9º Ano
<i>diffEm_1 até diffEm_24</i>	Representam 24 pontos de dificuldades que podem ser encontrados na assessoria do Ensino Médio
<i>assEm_1 até assEm_21</i>	Representam 21 pontos de assuntos que podem ser abordados na assessoria do Ensino Médio
<i>resultado</i>	Resultado da classificação do atendimento após agrupamento (ÓTIMO, REGULAR ou RUIM)

5) *Preparação para treinamento*: Nesta etapa se divide a base de dados entre teste e treino. Para isso, foi utilizado da biblioteca *caret*, o método *createDataPartition* [5]. Ele busca balancear de forma aleatória os dados com base no número de classes. Utilizou-se 80% da base para treino e 20% para teste, resultando em 568 e 140 registros respectivamente.

Por meio da função *trainControl* foi estipulado um *cross-validation* de *10-fold* para os dados de treino. Ou seja, estes dados são divididos em 10 partes que se alternam a cada rodada, uma sendo de treino e as outras nove para teste. Esta técnica utiliza a acurácia de cada uma das iterações para se extrair uma média geral. Obtendo assim, um melhor modelo que será treinado com todos os dados. [6]

6) *Treinamento do modelo e escolha dos parâmetros*: Nesta etapa, foi utilizado um algoritmo de Redes Neurais Artificiais com auxílio do pacote NNET [7] disponível para a linguagem R. Para a configuração do modelo, utilizando os hiperparâmetros disponíveis na biblioteca *caret*, buscou-se identificar qual seria a melhor quantidade de neurônios (*size*) e o melhor decaimento dos pesos (*decay*).

O *size* representa os neurônios da camada escondida que podem variar de 1 a N. Já o *decay*, representa a redução da taxa de aprendizagem, ou passo, com o objetivo de otimizar a descida de gradiente e se obter resultados mais precisos [8]. A faixa de valores para o *decay* pode ir de 0,1 a 0,9 com intervalo de 0,1 ou ainda menores, de 0,01 por exemplo. Com isso, é possível achar a melhor combinação de hiperparâmetros por meio da acurácia extraída a cada iteração.

O treinamento inicialmente foi realizado utilizando um *size* com intervalo de 1 a 10 e *decay* de 0,1. Com esses parâmetros já foi possível se obter ótimos resultados, contudo testes em intervalos maiores também foram realizados, mas não houve diferença nos resultados. A utilização de *decay* inferior a 0,1 se demonstrou computacionalmente custosa e demorada, também sem mudança no resultado. Por fim, o intervalo escolhido para o *size* foi de 1 a 3 e para o *decay*, de 0,1 a 0,3 com passo de 0,1.

Com o treinamento realizado, foi possível observar os resultados de cada hiperparâmetro representados de forma resumida na Tabela II. Foi possível obter uma acurácia de 99,11% por meio da utilização de 2 neurônios na camada interna e um *decay* de 0,1. Nota-se que o resultado se repete também com 3 neurônios, tanto com 0,1 ou 0,2 de *decay*. A variação da acurácia começa a aparecer a partir do passo de 0,3.

TABELA II
PRINCIPAIS RESULTADOS DO TREINAMENTO

<i>Size</i>	<i>Decay</i>	Acurácia
1	0.1	0.9525592
1	0.3	0.9454477
2	0.1	0.9911946
2	0.2	0.9894089
2	0.3	0.9858990
3	0.1	0.9911946
3	0.2	0.9894089
3	0.3	0.9858374

7) *Predição*: Nesta etapa, a predição foi realizada utilizando os dados de teste com 140 registros. Obteve-se como resultado a matriz de confusão representada na Tabela III. Com a utilização dos hiperparâmetros de *size* 2 e *decay* 0,1, a acurácia obtida foi de 97,86%. No entanto, a métrica utilizada para aprovação do modelo é a de especificidade, na qual o foco é detectar os negativos. Porque uma vez que errar a previsão do atendimento ruim para ótimo ou regular é mais danoso para o acompanhamento do projeto do que o contrário. Com isso, a especificidade do modelo alcançou 99,20% para a classe ruim, obtendo um nível aceitável segundo os coordenadores.

TABELA III
MATRIZ DE CONFUSÃO APÓS PREDIÇÃO

Predição	Referência		
	ÓTIMO	REGULAR	RUIM
ÓTIMO	102	1	0
REGULAR	1	20	0
RUIM	0	1	15

8) *Construção do software*: Por fim, com o modelo de RNA para ser utilizado, se realizou o desenvolvimento da aplicação. Ela consiste em uma listagem de todas as escolas, contendo a quantidade de atendimentos classificados pelo modelo IA. Junto a isto, uma indicação visual, em formato de círculo colorido é atribuído a instituição, com base na média da qualidade dos atendimentos. Indicando de forma clara ao usuário a situação geral da escola.

O fluxo do funcionamento do software está descrito pela Figura 1. Os registros de atendimentos, assim que são inseridos no software interno da ZOOM Education, também são enviados ao software de IA. Ao receber e realizar o pré-processamento, realiza a predição com a RNA e atualiza, com o resultado da classificação, as métricas da qualidade dos atendimentos.

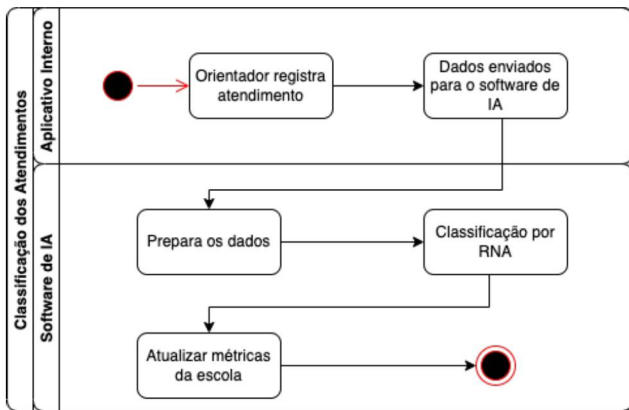


Fig. 1. Fluxo de funcionamento do Software de IA

C. Tecnologias

O software foi desenvolvido em um MacBook Pro com processador i7 de 4-núcleos a 2,2 GHz, com sistema operacional Mac OS 12.4. Foram utilizadas as seguintes aplicações e bibliotecas:

- Linguagem R, versão 4.1.0;
- RStudio, versão 2022.02.2;
- Pacote *caret*, versão 6.0-92;
- Pacote *mice*, versão 3.14.0;

II. RESULTADOS E DISCUSSÕES

Para o desenvolvimento do software foram criados alguns artefatos que servem também como documentação. Entre eles está o diagrama de atividades, mostrando a ordem dos acontecimentos dentro do programa. Há também o diagrama físico do banco de dados, explorando os atributos utilizados pelo software em seu funcionamento. Por fim, um protótipo da tela do aplicativo representando a visão do usuário em produção.

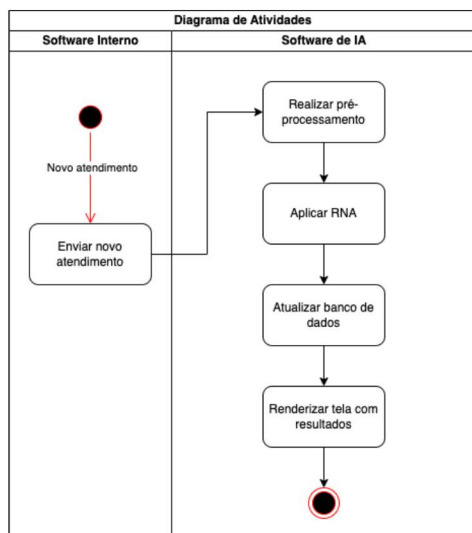


Fig. 2. Diagrama de Atividades

A sequência do funcionamento do software pode ser observada na Figura 2. Neste tipo de diagrama, pode-se observar graficamente o comportamento e a atuação no quesito do negócio onde está inserido. Esta sequência lógica pode se demonstrar fundamental no desenvolvimento ágil do software, uma vez que facilita o entendimento das ideias. [9] O gatilho inicial é a entrada de um novo registro de assessoria realizado pelo orientador da escola. Ele é realizado por meio do aplicativo próprio da empresa. Esse registro é enviado ao software no qual é tratado realizando a mesma limpeza e adequações necessárias descritas na etapa de pré-processamento. Com os dados já ajustados, eles passam pelo modelo RNA, desenvolvido ao longo desse trabalho, para classificação do atendimento em questão. Com isso, o banco de dados é atualizado com as novas métricas e a tela, com a listagem de classificações, pode ser atualizada com as novas informações.

Software	
id: INT	[PK]
id_escola: INT	
nome_escola: VARCHAR(45)	
otimo: INT	
regular: INT	
ruim: INT	
media: VARCHAR(45)	

Fig. 3. Diagrama Físico de Dados

O banco de dados do software é bem simplificado, contendo apenas os resultados das previsões e os identificadores das escolas. O diagrama físico está representado pela Figura 3. Esse tipo de modelagem está presente no final do processo de design do banco de dados, pois instrui como serão registradas as informações. Além disso, é o detalhamento das tabelas que inicialmente eram classes. [10]. Nele é possível observar um identificador padrão como chave primária de cada registro. Há o identificador da escola utilizado pelo sistema interno da empresa, para que o software consiga relacionar o novo registro com o resultado que será salvo (garantindo que seja da mesma escola).

Cada uma das três métricas de classificação dos atendimentos da escola são armazenados de forma numérica. Ou seja, ao final da previsão de um atendimento, sua quantidade é incrementada conforme o resultado predito. Já a média geral, é recalculada logo após a classificação do atendimento. Salvando apenas a classificação em formato de texto entre ótimo, regular ou ruim.



Fig. 4. Protótipo da Tela do Software

A fim de demonstrar de forma gráfica a aparência do software para o usuário, temos representado pela Figura 4, o protótipo da tela. Nela é possível observar a lista das escolas presentes na base da empresa e suas métricas de atendimentos. Observa-se ao lado esquerdo do nome, a média geral da escola, facilitando a tomada de decisão do coordenador para com esse cliente. Já do lado direito do nome da instituição, está a régua que traz a quantidade de cada uma das classificações.

III. CONSIDERAÇÕES FINAIS

A base de dados utilizada nesse estudo pode ser considerada pequena e desbalanceada, mas leva em conta dados reais obtidos em atendimentos. A grande diferença na quantidade de registros classificados como atendimentos ótimos ou ruins deixa evidente essa questão do desbalanceamento. Com isso, existe a possibilidade de gerar algum viés nos algoritmos de classificação. Há ainda, a possibilidade de treinamentos futuros com bases de dados maiores para se obter modelos mais precisos.

O software demonstra ter grande potencial em seu objetivo, uma vez que analisar os atendimentos prestados aos clientes, ações corretivas poderão ser tomadas a fim de garantir maior qualidade do projeto. O registro de atendimentos é uma prática já muito comum dentro da organização atual, e parece ser recorrente nesse ramo. Mas a interpretação desses dados, ou análise mais profunda, ainda é

um campo pouco explorado. Com esse software, pode-se obter de forma simples e objetiva, um panorama mais sistêmico de como está o desempenho dos programas de educação tecnológica. Fornecendo assim, embasamento aos coordenadores para o planejamento de novas ações.

REFERÊNCIAS

- [1] ZOOM BLOG. O Uso da Tecnologia no estudo de conteúdos curriculares. 2017. Não paginado. Disponível em: <http://blog.zoom.education/2017/12/o-uso-da-tecnologia-no-estudo-de-conteudos-curriculares/>. Acesso em: 25 de julho de 2022.
- [2] BRASIL. Ministério da Educação. Base Nacional Comum Curricular. Brasília, 2018. Disponível em: <http://basenacionalcomum.mec.gov.br/abase/#fundamental>
- [3] DATA CAMP. RDocumentation: data.frame: Data Frames. 3.6.2. 2019. Disponível em: <https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/data.frame>. Acesso em: 22 ago. 2022.
- [4] Imaster1. Data Mining dna prática: Algoritmo K-Means. DevMedia, 2007. Disponível em: <https://www.devmedia.com.br/data-mining-na-pratica-algoritmo-k-means/4584>. Acesso em 25 de julho de 2022
- [5] DATA CAMP. RDocumentation: createDataPartition: Data Splitting functions. 6.0. 2019. Disponível em: <https://www.rdocumentation.org/packages/caret/versions/6.0-92/topics/createDataPartition>. Acesso em: 22 ago. 2022.
- [6] RABELLO, Eduardo. Cross Validation: Avaliando seu modelo de Machine Learning. Medium, 7 jul. 2019. Disponível em: <https://medium.com/@edubrazabello/cross-validation-avaliando-seu-modelo-de-machine-learning-1fb70df15b78>. Acesso em: 23 ago. 2022.
- [7] DATA CAMP. RDocumentation: nnet: Fit Neural Networks. 7.3. 2019. Disponível em: <https://www.rdocumentation.org/packages/nnet/versions/7.3-17/topics/nnet>. Acesso em: 22 ago. 2022.
- [8] DATA SCIENCE ACADEMY (Brasil). Aprendizado Com a Descida do Gradiente: Descida do Gradiente. In: DEEP Learning Book. cap. 12, p. 1-1. Disponível em: <https://www.deeplearningbook.com.br/aprendizado-com-a-descida-do-gradiente/>. Acesso em: 23 ago. 2022.
- [9] VENTURA, Plínio. Entendendo o Diagrama de Atividades da UML. Até o Momento, 2016. Disponível em: <https://www.ateomomento.com.br/uml-diagrama-de-atividades/>. Acesso em: 5 dez. 2022.
- [10] UFPE (Brasil). Diretriz: Modelo de Dados. In: Diretriz: Modelo de Dados. 2008. Disponível em: https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/guidelines/data_model_80FB2539.html. Acesso em: 26 out. 2022.