Classificação de Textos Escolares com Aprendizado de Máquina

Nelson J. Dressler
nelsondr58@gmail.com
Universidade Tecnológica Federal do Paraná (UTFPR)
Dois Vizinhos, Paraná, Brasil

Alinne C. Correa Souza
alinnesouza@utfpr.edu.br
Universidade Tecnológica Federal do Paraná (UTFPR)
Dois Vizinhos, Paraná, Brasil

Lincoln Magalhães Costa
costa@cos.ufrj.br
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, Rio de Janeiro, Brasil

Francisco Carlos M. Souza
franciscosouza@utfpr.edu.br
Universidade Tecnológica Federal do Paraná (UTFPR)
Dois Vizinhos, Paraná, Brasil

Rafael G. Mantovani
rafaelmantovani@utfpr.edu.br
Universidade Tecnológica Federal do Paraná (UTFPR)
Apucarana, Paraná, Brasil

RESUMO

Nowadays, during the teaching-learning process, it is prevalent for students to seek information about the subjects to complement studies or execute school work. For this, it is possible to use search engines on the Internet, but the texts do not always have the complexity compatible with the student's level of education. A solution for this purpose is the automatic filtering of texts, where computational models are created for textual classification according to their contexts. Many studies have presented solutions based on Machine Learning techniques, but there is no comparison nor an analysis considering these methods. This work explores this scenario throughout implementing machine learning algorithms. In the solution, Natural Language Processing techniques are applied in order to select characteristics of the textual contents and introduce them into the classification pipeline. Experiments were conducted and the results obtained suggest that, machine learning algorithms have a good predictive performance values, with class-balanced accuracy of around 0.89 in the test data.

KEYWORDS

Educational texts classification, Textual complexity, Natural language processing, Machine learning

1 INTRODUÇÃO

Durante o processo de ensino-aprendizagem, muitos alunos costumam complementar seus estudos acerca das disciplinas ministradas, bem como necessitam de acesso a materiais específicos para a realização de trabalhos escolares. Para tal, é possível alcançar a informação desejada por meio de diversos meios de comunicação, tais como: livros, jornais, revistas e, principalmente, a *Internet*, a partir de seus motores de busca.

Entretanto, os materiais encontrados estão sujeitos a não refletir com exatidão o tema e/ou grau de escolaridade do aluno: podem não estar relacionados ou ainda acima ou abaixo de suas capacidades, mesmo compatível com o tema em questão. Consequentemente, o estudante acaba tendo a necessidade de filtrar os dados textuais recuperados manualmente, a fim de garantir que os mesmos referenciem de fato o tema de estudo, assim como o nível de complexidade esperado. Isto torna a tarefa de busca de conteúdo árdua e impede que o prazo de conclusão seja estimado com antecedência.

Uma possível solução de suporte à filtragem correta de conteúdos textuais encontrados na *Internet* é por meio de processamento computacional e análise descritiva dos dados, de modo que seja possível classificar ou rotular automaticamente a complexidade textual para os diversos estágios escolares. Neste sentido, visando o processamento computacional sobre dados textuais, é possível utilizar técnicas e conceitos de **Processamento de Linguagem Natural (PLN)**, uma área interdisciplinar situada na Computação, e que faz o uso tanto de técnicas de **Inteligência Artificial (IA)** quanto conceitos de **Linguística** [1]. O objetivo é desenvolver ferramentas que permitam manipular, descrever e até mesmo classificar conteúdos textuais, de modo que a máquina possa compreender a linguagem humana.

Muitos estudos que exploram PLN para a classificação de textos apresentam uma sequência de tarefas completa (*pipeline*), compreendendo as etapas de: coleta dos dados, extração de características, e a classificação propriamente dita. O estado da arte em classificação textual tem explorado algoritmos de **Aprendizado de Máquina** (AM) [2–4], sejam abordagens baseadas em algoritmos tradicionais ou em *Deep Learning* (DL), devido à robustez destes modelos preditivos.

Desta forma, este trabalho tem por objetivo explorar o uso de algoritmos de AM para a classificação de textos escolares de acordo com suas respectivas disciplinas. Além disso, o uso de algoritmos tradicionais de AM possibilita automatizar a tarefa de classificação dos dados textuais, visando a automatização completa do *pipeline*.

O presente trabalho está organizado da seguinte forma: as Seções 2 e 3 apresentam os fundamentos teóricos envolvidos na elaboração do trabalho e uma breve descrição dos trabalhos relacionados, respectivamente; a Seção 4 descreve a metodologia experimental necessária para realização e reprodução dos experimentos; os resultados e discussões são apresentados na Seção 5; e, por último, são apresentadas as conclusões e possíveis caminhos para os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Os conceitos teóricos empregados no desenvolvimento deste trabalho são apresentados a seguir.

2.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) [5] é uma área de pesquisa que explora técnicas de representação e manipulação de

textos com o propósito de criar modelos computacionais capazes de abstrair tarefas como leitura e escrita, simulando a linguagem humana. Dentre estas tarefas, é possível citar a classificação de textos, sumarização, análise de sentimentos, tradução automática, entre outras.

Durante a elaboração de uma solução baseada no processamento de textos, existem diferentes técnicas que podem ser exploradas a fim de se extrair características ou remover informação que não seja útil ao domínio. Esse processo é comumente referido como pré-processamento textual. Na prática, existem diversos métodos de pré-processamento de textos [2] e seleção de características, os mais comuns e que valem ser destacados são:

- **Remoção de ruídos:** remove "ruídos", dados textuais que não possuem relevância informativa, como espaços adicionais, pontuação, acento ou demais caracteres especiais irrelevantes à análise textual;
- **Remoção de palavras vazias (*stopwords*):** consiste na remoção de termos sem representatividade semântica no texto, como artigos e preposições;
- **Stemização (*stemming*):** consiste no processo de redução das palavras ao seu radical, desconsiderando sufixos;
- **Vetorização:** consiste em organizar todos os termos presentes no *corpus* (conjunto de documentos) como um vetor e atribuir valores numéricos a cada posição, que as frequências absolutas de cada termo no documento;
- **TF-IDF:** consiste em atribuir pesos conforme a frequência de um termo num documento específico (TF) e, ao mesmo tempo, ponderar negativamente com a frequência alta do mesmo termo em diversos documentos (IDF), atribuindo um peso menor no documento corrente;
- **Seleção de características:** consiste na redução de dimensionalidade do espaço de atributos de entrada, a partir da seleção das características que melhor representem os dados na sua totalidade. Neste contexto, os textos (entradas) são associados às classes (saídas) e, com base em testes estatísticos univariados são atribuídas pontuações para cada termo, representando uma característica específica. Como consequência, estes termos são filtrados e ranqueados os K atributos mais relevantes no *dataset* completo [6]; e
- **Discretização ou Binarização:** consiste em converter valores contínuos em inteiros que representem estratos de dados, onde cada grupo possui um significado ou valor categórico.

2.2 Aprendizado de Máquina

O Aprendizado de Máquina (AM) [7], é uma subárea da IA cuja característica intrínseca é a construção de modelos matemáticos que visam mapear e identificar padrões nos dados. Isso é feito pela indução de modelos, tanto supervisionados como não-supervisionados, para resolver problemas práticos e extrair conhecimento útil do domínio estudado. Modelos supervisionados são gerados por algoritmos que trabalham com dados rotulados, modelados em tarefas de classificação ou regressão, enquanto modelos não-supervisionados são gerados por meio de algoritmos de aprendizado não-supervisionado (agrupamento de dados). Com base no modelo gerado, é possível criar um sistema automatizado, viabilizando a indução e a conclusão sobre dados desconhecidos ou futuros.

Considerando a manipulação de dados rotulados, a indução dos modelos preditivos pode ocorrer através de diversos métodos [8], seguindo diferentes vieses de aprendizado, e organizados em diferentes paradigmas. Alguns deles são:

- **Métodos baseados em distâncias:** realizam predições com base na noção de "proximidade" dos dados. Novos exemplos são classificados de acordo com sua vizinhança no espaço de entrada. Neste paradigma, é possível citar o algoritmo dos K vizinhos mais próximos (*k-Nearest Neighbors* - k-NN);
- **Métodos probabilísticos:** realizam predições com base no Teorema de Bayes, onde é avaliada a probabilidade de ocorrência de um evento (classe), dado um conjunto de valores do conjunto de entrada (atributos). Neste paradigma, é possível citar os algoritmos Naïve Bayes (NB) com distribuição de Bernoulli ou Multinomial;
- **Métodos baseados em procura:** realizam predições com base na exploração de um espaço de possíveis soluções. Neste paradigma, é possível citar as Árvores de Decisão (*Decision Trees* - DT), que a cada iteração realiza cortes ortogonais ao espaço do domínio, sempre selecionando o melhor atributo que reduz ao máximo a incerteza na identificação dos padrões;
- **Métodos baseados em otimização:** realizam predições com base na otimização de alguma função custo, seja o objetivo minimizá-la ou maximizá-la. Neste paradigma, é possível citar os algoritmos de Gradiente Descendente Estocástico (*Stochastic Gradient Descendent* - SGD), Regressão Logística (RL), Máquinas de Vetores de Suporte (*Support Vector Machines* - SVMs) e Redes Neurais Multicamadas (*Multilayer Perceptron* - MLP); e
- **Métodos baseados em comitês (*ensembles*):** realizam predições com base na combinação e diversificação de uma série de classificadores fracos e na diversificação da amostragem dos dados. A amostragem dos dados pode ocorrer tanto nas instâncias como nos atributos, de modo que, ao final, haja uma decisão consensual entre os classificadores (comitê). Neste paradigma, é possível citar a Floresta Aleatória (*Random Forest* - RF).

3 TRABALHOS RELACIONADOS

Esta seção apresenta uma síntese dos trabalhos que motivaram e apresentaram soluções compatíveis com o problema levantado neste trabalho, os quais estão relacionados com tarefas de classificação de textos.

Em Zanchini [2], o autor criou um classificador de textos extraídos do `Twitter` com o intuito de identificar dados depressivos. Para tal, foram utilizadas SVMs e uma etapa anterior de pré-processamento dos *tweets* com o objetivo de eliminar informações irrelevantes. Como resultados obtidos, houve uma taxa de acerto de 99,7% sobre os dados de treinamento, realizando também Busca em Grade (*Grid Search* - GS) para obtenção das melhores combinações de hiperparâmetros.

Já em Silva [3], o autor propõe um classificador de textos referentes a crimes contra a honra, conforme a configuração do tipo de crime, utilizando DL e PLN. Para tal, foram utilizadas as seguintes categorias: ameaça, calúnia, difamação, injúria e injúria racial; e alguns tipos de redes neurais, como: Redes NeuraisPprofundas (*Deep Neural Networks* - DNNs), MLPs, Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNNs), Redes Neurais Recorrentes (*Recurrent Neural Networks* - RNNs), entre outros. Os melhores resultados foram obtidos usando-se DNNs e etapas de pré-processamento textual.

3.1 Classificação de Textos Educacionais

Dentre os estudos identificados na literatura mais relacionados a este estudo foram selecionados os seguintes artigos: Ferreira-Mello et al. [9], Gazzola et al. [10], Sha et al. [11], Wang et al. [12], todos descritos nos próximos parágrafos.

No trabalho de Ferreira-Mello et al. [9] é apresentada uma visão geral da área de Mineração de Texto Educacional, na qual foram identificadas as técnicas de mineração de texto mais utilizadas, os recursos educacionais mais utilizados e as principais aplicações ou objetivos educacionais. Neste contexto, foi conduzida uma revisão sistemática no período de Janeiro de 2016 a Julho de 2018 nas bases *IEEE Xplore, Springer, ScienceDirect, ACM e Google Scholar*, identificando um total de 343 artigos. De acordo com os resultados alcançados foi possível identificar que as principais técnicas de mineração de texto são a classificação de texto e o processamento de linguagem natural; os principais recursos educacionais são fóruns, trabalhos online e ensaios, porém também é importante destacar um interesse crescente por outras fontes como as redes sociais com a criação de aplicativos de comunidades; e o principal objetivo educacional consiste no auxílio dos instrutores na avaliação do desempenho dos alunos.

Wang et al. [12] apresenta um método de classificação de notícias educacionais com base no dicionário emocional afim de auxiliar os departamentos de educação na identificação rapidamente as últimas notícias educacionais e fornecer suporte de aplicativos para operação diária e gerenciamento de educação. Para isso, os autores inicialmente resumem o dicionário de sentimento comum e as palavras de sentimento relacionadas à educação manualmente. Em seguida, é utilizado o algoritmo SO-PMI suavizado de Laplace para expandir o dicionário emocional para notícias educacionais. Por fim, toda a notícia é dividida em várias cláusulas, e o dicionário de emoções é usado para calcular as tendências afetivas da notícia educacional, e então a notícia educacional é classificada. Para avaliar o método proposto foi conduzido um experimento que utiliza 5.000 notícias educacionais como dados experimentais a fim de comparar o efeito de dois dicionários emocionais diferentes no método de classificação de textos de notícias educacionais. De acordo com os resultados alcançados, o método de classificação de notícias educacionais com base no dicionário emocional é eficaz e viável. No entanto, alguns pontos relacionados ao manuseio de palavras ambíguas e a conversão de frases ainda precisam ser melhorados.

Em um trabalho recente, Gazzola et al. [10] analisam a complexidade do texto dos Recursos Educacionais Abertos (*Open Educational Resources* - OER) no idioma português do Brasil. Para isso, utilizam uma abordagem de aprendizado multitarefa (*Multitask Learning* - MTL) em um corpus recentemente criado de narrativas faladas transcritas produzidas por alunos da quarta série para alunos do 1° ano do ensino médio. O resultado do modelo MTL com duas tarefas auxiliares apresenta uma medida F-score de 0.955, uma melhoria de 0.15 pontos em relação aos resultados anteriores.

O trabalho de Sha et al. [11] aborda um dos principais problemas envolvendo modelos inteligentes: o viés causado por dados desbalanceados e desiguais. Assim, o estudo teve como objetivo quantificar a consciência que um Modelo de Linguagem Pré-Treinado (PLM, mais especificamente BERT) tem em relação aos atributos protegidos das pessoas e aumentar o BERT para melhorar a equidade de predição dos modelos, inibindo essa consciência. Nos experimentos, em comparação com uma linha de base sem qualquer pré-formação adicional, o método proposto melhorou não apenas a equidade (com uma melhoria máxima de 52,33%), mas também a precisão (com uma melhoria máxima de 2,53%). Segundo os autores, o método também pode ser generalizado para qualquer PLM e atributos demográficos.

3.2 Considerações gerais

É importante salientar que alguns dos trabalhos relacionados abordam tarefas de classificação de textos de modo geral, não se atendo apenas a textos didáticos, como explora este trabalho. Entretanto, os experimentos realizados poderiam também ser aplicados na resolução deste estudo e, por isso, foram citados como parte dos trabalhos relacionados.

4 METODOLOGIA EXPERIMENTAL

Nesta seção são descritas todas as etapas da elaboração da solução do problema de pesquisa investigado.

4.1 Conjunto de dados (*dataset*)

Para a realização dos experimentos foi utilizado um conjunto de dados (*dataset*) denominado "Corpus de Complexidade Textual para Estágios Escolares do Sistema Educacional Brasileiro" [13] [1]. O *dataset* é composto por textos didáticos rotulados de acordo com a complexidade textual, correspondendo a estágios escolares conforme rege o Sistema Educacional Brasileiro (LDB) [14].

Este *dataset* descreve um problema de classificação multiclasse com quatro categorias, que definem o tipo de conteúdo de um texto em diferentes níveis de escolaridade: Ensino Fundamental 1, Ensino Fundamental 2, Ensino Médio e Ensino Superior. Este problema é também um problema desbalanceado, como destacado pela Tabela 1: existem 826 exemplos de textos de Ensino Superior (39,79%), 628 exemplos de textos de Ensino Médio (30,25%), 325 exemplos de textos de Ensino Fundamental 2 (15,66%) e 297 exemplos de textos de Ensino Fundamental 1 (14,31%). Portanto, existe também uma dificuldade inerente ao problema pela quantidade de textos de cada tipo de escolaridade.

4.2 Algoritmos de AM

Nos experimentos de classificação textual foram testados diferentes algoritmos de AM: k-Vizinhos Mais Próximos (K-NN), Regressão Logística (RL), Gradiente Descendente Estocástico (SGD), Naïve Bayes (NB) Bernoulli, Naïve Bayes (NB) Multinomial, Máquinas

[1]Este *dataset* é público e pode ser acessado pelo seguinte link: https://github.com/gazzola/corpus_readability_nlp_portuguese.

Tabela 1: Distribuição de classes e porcentagem de exemplos em relação ao conjunto total.

Classe	N#	%
Ensino Fundamental 1	297	14,31
Ensino Fundamental 2	325	15,66
Ensino Médio	628	30,24
Ensino Superior	826	39,79
Total	2076	100,00

de Vetores de Suporte (SVM), Árvore de Decisão (DT), Floresta Aleatória (RF) e Perceptron Multicamadas (MLP). A escolha destes algoritmos deu-se por serem os algoritmos explorados pelos trabalhos relacionados, e por possuírem diferentes viéses indutivos, ou seja, diferentes formas de aprender com os dados [7].

4.3 *Pipeline* Implementado

Nos experimentos, foi utilizado um *pipeline* convencional comum à tarefas de classificação [15], compreendendo as seguintes etapas: análise exploratória de dados (*Exploratory Data Analysis* - EDA); pré-processamento; seleção e indução de modelos, incluindo o ajuste de hiperparâmetros; validação dos modelos induzidos e avaliação das predições.

(1) **Pré-Processamento:** consistiu nas etapas de uniformização textual; descarte de ruídos e palavras vazias; stemização; vetorização; obtenção de pesos baseados em TF-IDF; e conversão em matrizes numéricas densas[2];
(2) **Seleção de Modelos:** os modelos foram induzidos e avaliados por meio de uma estratégia de validação cruzada (*Cross-Validation*- CV) aninhada, combinando amostragem por *holdout* com CV. O laço externo divide os dados usando *holdout*, com 80% dos dados selecionados para treinamento dos modelos e 20% para teste. Internamente é usada uma divisão por CV para treinamento e ajuste de hiperparâmetros dos algoritmos de classificação. Além disso, a CV é executada com 10 partições, as quais foram usadas para indução dos modelos pelos algoritmos descritos na Seção 4.2.
(3) **Ajuste de Hiperparâmetros** (HPs): os hiperparâmetros dos principais algoritmos de classificação e das funções de pré-processamento também foram ajustados internamente, incluindo a seleção das melhores características com base em testes estatísticos. O processo foi realizado por meio da Busca Aleatória (*Random Search* - RS) com um *budget* de 100 avaliações por partição da CV interna. É importante observar que foi utilizada a Acurácia Balanceada por Classes (*Balanced per Class Accuracy) - BAC* para avaliação dos melhores classificadores e hiperparâmetros, devido ao desbalanceamento das classes.

Uma vez executado todo o *pipeline*, os resultados foram então também avaliados e mensurados em termos da Acurácia Balanceada por Classes (BAC) [16], além de outras métricas de classificação, obtidas por meio da matriz de confusão dos modelos gerados: Precisão, Revocação e medida F1 (F-Score). Ademais, outra forma de mensuração adotada neste trabalho foi a comparação dos resultados dos modelos obtidos com três *baselines*: o primeiro composto por predições aleatórias simples; o segundo pela predição da classe majoritária; e o terceiro pela predição da classe minoritária.

4.4 Validação estatística

Para consolidação dos resultados obtidos em ambas as soluções, foi realizada uma etapa de validação estatística e comparação dos modelos induzidos. Todos os modelos tiveram suas acurácias parciais comparadas, dois a dois, por meio do teste não paramétrico de Wilcoxon, com $\alpha = 0.05$ [17]. Este teste permite avaliar uma possível equivalência de distribuições de performances e, assim indicar diferenças estatísticas no desempenho de cada par de classificadores.

4.5 Reprodutibilidade dos Experimentos

Para a codificação dos experimentos, foi utilizada a linguagem *Python*, na versão 3.7.1. Foi usado também o pacote de bibliotecas do *Anaconda*, versão 4.10.1, com as seguintes bibliotecas: *Numpy* [18], *Pandas* [19], *Matplotlib* [20], *Seaborn* [21], *Graphviz* [22], *Scikit-learn* [23], *NLTK* [24]. Para reprodutibilidade dos experimentos foi adotado um valor de semente aleatória (*seed*) igual a 42[3].

5 RESULTADOS E DISCUSSÕES

Os principais resultados obtidos nos experimentos são descritos a seguir.

5.1 Pré-processamento adicional do dataset

Durante as etapas de EDA e pré-processamento, mais especificamente após a limpeza inicial dos dados (uniformização, descarte de ruídos e de palavras vazias padrão[4]), foram geradas inicialmente quatro nuvens de palavras, uma para cada classe, que correspondem a cada grau de escolaridade.

A partir desta análise inicial foram identificados termos irrelevantes, além das *stopwords*, que apresentavam uma alta frequência porém sem adicionar informação relevante ao contexto do problema. Por exemplo, os termos '`pode`', '`voce`' e '`sim`' são palavras sem contexto nos textos, e não descrevem nenhum tipo de relação ao conteúdo descrito. Assim, foi necessária uma segunda etapa de limpeza, removendo também estes "termos irrelvantes". [5] Após a remoção das palavras vazias adicionais, foram geradas novas nuvens de palavras, ilustradas na Figura 1. Comparando com as nuvens anteriores, é possível notar visualmente nas nuvens o efeito da remoção destes termos irrelevantes.

Uma característica interessante quando analisamos a Figura 1 é que alguns termos possuem maior frequência em uma classe textual específica. Por exemplo: as palavras '`brasil`', '`agua`',

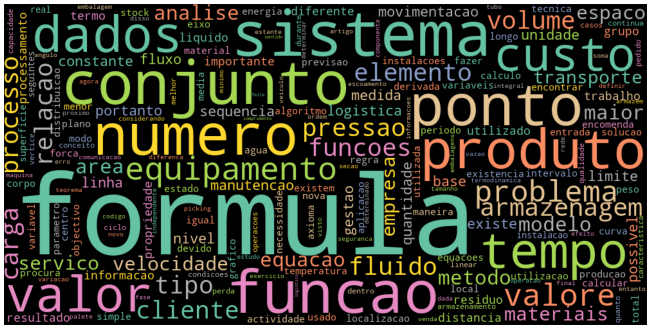
[2]Por padrão, a biblioteca *Numpy* do *Python* condensa matrizes com grandes quantidades de zeros para economizar espaço de memória. Esta conversão visa expandir a matriz para viabilizar as operações posteriores.

[3]O código completo desenvolvido pode ser acessado no link: https://github.com/nelsondressler/ecd_utfpr_dv_tcc/blob/main/TCC_Text_Classification_hcv.ipynb
[4]Relacionadas pela biblioteca de PLN do *Python* NLTK
[5]A listagem completa de todas as palavras irrelevantes identificadas pode ser encontrada no código-fonte da aplicação.

(a) Ensino Fundamental 1.

(b) Ensino Fundamental 2.

(c) Ensino Médio.



(d) Ensino Superior.

Figura 1: Nuvens de palavras dos textos analisados após remoção de palavras irrelevantes. O tamanho da palavra é proporcional à frequência relativa nos textos.

'crianca' e 'historia' aparecem com maior destaque em textos de conteúdo do Ensino Fundamental 1. Já nos textos do Ensino Fundamental 2, as palavras que se destacam são 'problema', 'vida', 'trabalho' e 'social'. Textos do Ensino Médio já ressaltam termos como 'celula', 'estado' e 'cidade'; enquanto conteúdos do Ensino Superior destacam palavras como 'sistema', 'conjunto', 'dados' e 'funcao'.

5.2 Desempenhos gerais dos algoritmos

O desempenho geral de cada algoritmo de classificação, utilizando CV sobre os dados de treinamento, é apresentado na Tabela 2. Para cada algoritmo são apresentados: o valor correspondente de BAC média obtido no conjunto de treinamento (via CV), e valor de BAC no conjunto de teste (via *holdout*).

A partir dos resultados apresentados na tabela é possível evidenciar algumas informações sobre os modelos induzidos, dividindo-os em quatro grupos distintos:

- **Grupo A:** obtiveram os melhores resultados no conjunto de teste. Esse grupo é composto pelos algoritmos MLP (0.88) e SGD (0.887);
- **Grupo B:** formam um segundo grupo de soluções os algoritmos SVM, RL e K-NN, com valores de desempenho no conjunto de testes entre [0.80, 0.86].
- **Grupo C:** os demais algoritmos de AM, com exceção dos baselines, obtiveram valores de BAC entre [0.63, 0.75]. Apesar deste grupo compreender um intervalo maior quando comparado aos primeiros dois, é possível afirmar que, ao menos, houve aprendizado a partir dos dados; e
- **Grupo D:** composto pelos *baselines* de classes majoritária e minoritária (0.25), e aleatório simples (0.2366).

Tabela 2: Desempenho dos classificadores em termos da BAC nos conjuntos de treino e teste. Os melhores resultados obtidos em cada partição estão destacados em negrito.

Grupo	Algoritmo	BAC Treinamento	BAC Teste
A	MLP	**0.9034**	0.8804
	SGD	0.9006	**0.8876**
B	SVM	0.8775	0.8623
	RL	0.8661	0.8381
	K-NN	0.8221	0.8039
C	NB-Bern.	0.7868	0.7565
	RF	0.7754	0.7865
	DT	0.6891	0.6503
	NB-Multi.	0.6252	0.6321
D	Major.	0.2500	0.2500
	Minor.	0.2500	0.2500
	Random	0.2367	0.2636

Ainda sobre os valores médios de BAC, ao analisar o ranking dos classificadores, é possível inferir um relacionamento entre a

estrutura dos dados e da natureza do problema com os viéses indutivos dos modelos que obtiveram desempenhos bons ou ruins. Primeiramente, é possível perceber um paralelo entre os melhores modelos induzidos (MLP e SGD), pois ambos possuem um viés de otimização baseado na função de custo de gradiente descendente estocástico.

Além disso, os quatro melhores modelos em desempenho (MLP, SGD, SVM, RL) são algoritmos que manipulam apenas informações numéricas, o que é justificável, graças a etapa anterior de pré-processamento, onde são obtidos valores numéricos de pesos para cada termo nos documentos do *dataset*. Ainda analisando com base no desempenho do RL, é possível sugerir um grau de linearidade entre as classes do problema.

Em contrapartida, observando os resultados obtidos pela DT e RF, é possível perceber que os dados não se comportam bem em algoritmos que realizam cortes ortogonais nos dados, mesmo num cenário onde evitam o sobre-ajuste com a utilização de comitês.

A fim validar estatisticamente a significância dos resultados obtidos, foi aplicado o teste não paramétrico de Wilcoxon pareado com $\alpha = 0.05$. Os testes foram aplicados realizando-se 10 repetições do processo de treinamento variando o *seed* gerador das partições de treino e teste. A Figura 2 apresenta estes resultados, destacando os pares sem diferenças significativas, com P-Valores maiores que 0.05, na cor clara, e os demais na cor escura.

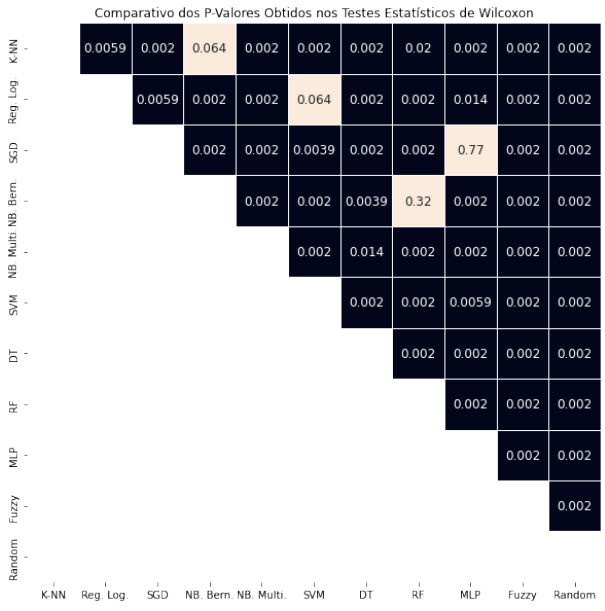


Figura 2: Teste de Wilcoxon pareado comparando os algoritmos de AM com $\alpha = 0.05$

Os resultados dos testes estatísticos mostram que apenas alguns pares de modelos não apresentam diferenças significativas (células claras), como os pares de modelos: MLP e SGD, SVM e RL, K-NN e NB Bernoulli, e NB Bernoulli e RF. Com base nos modelos equivalentes identificados, é possível observar que estes pares de modelos correspondem às proximidades diretas nas colocações de BAC média. Já em relação aos *baselines* todos os algoritmos foram estatisticamente superiores, o que é bom e desejado, caso contrário não estaria acontecendo nenhum tipo de aprendizado no problema.

5.3 Análise das Predições dos Modelos

A Figura 3 apresenta as matrizes de confusão dos melhores modelos obtidos (SGD e MLP). Nas figuras, é possível perceber que ambas as soluções tiveram desempenho similar para identificar a classe majoritária, que representam os textos com conteúdo do Ensino Superior. Há também uma frequência similar em classificações errônea de textos de Ensino Superior atribuídos à classe de Ensino Médio. Isso pode ter ocorrido devido a termos similares contidos na descrição dos exemplos de ambas as classes, mesmo que não ocorrendo em uma mesma frequência.

A diferença de desempenho ocorre na predição das demais classes: conteúdos textuais de Ensino Médio e Fundamental 1. O SGD faz confusão entre alguns textos das classes Ensino Fundamental 1 e 2. Já o modelo da MLP, tem a maioria dos seus erros misturando equivocadamente também alguns textos com conteúdo de Ensino Médio. Nesse sentido, os modelos não conseguem discernir exemplos que podem ser dúbios, carecendo de um maior enriquecimento na geração das características ou ajuste finos dos hiperparâmetros dos modelos, alternativas frequentemente relatadas na literatura.

6 CONCLUSÕES

Neste artigo, foi investigado o uso de algoritmos de AM para uma tarefa de classificação de textos escolares. A partir dos experimentos realizados pode-se observar que os melhores resultados foram obtidos pelos algoritmos SGD e MLP, sem diferença estatística entre os mesmos. Os resultados mostram que a aplicação de técnicas de pré-processamento, incluindo termos irrelevantes para o contexto do problema, e o processo de stemização refinaram e aperfeiçoaram a generalização dos modelos indutivos. Estes processos "homogenizaram" a distribuição dos termos descritivos de cada classe, eliminando ruídos que poderiam gerar falsas predições.

Além disso, ao comparar os principais modelos de AM, pode-se perceber que, mesmo com a execução de um ajuste de HPs simples, e testando diferentes viéses indutivos, os algoritmos puderam ser agrupados de acordo com o desempenho preditivo. Algoritmos inerentes a dados numéricos tiveram desempenho superior a algoritmos mais simples, ou baseados em árvores e probabilidade. Isso reflete também o processo de extração de características da etapa de pré-processamento, beneficiando estes algoritmos em questão.

Por fim, como trabalhos futuros propõe-se refinar os métodos de seleção de modelos, utilizando validação cruzada aninhada com duas etapas de CV e uma etapa mais robusta para geração de características descritivos do dataset. A geração de um *voting* usando os melhores algoritmos também poderia tornar as predições mais robustas. Outro ponto passível de melhora é na própria modelagem do problema. A classificação de textos foi realizada apenas comparando-se as classes um-contra-um e não um-contra-o-resto, método no qual poderia oferecer um desempenho maior com múltiplos modelos binários. Existem também diversos hiperparâmetros que podem ser ajustados, e por conta disso, poderiam ser utilizas técnicas de ajuste de hiperparâmetros mais robustas, como Otimização de Enxame de Partículas (*Particle Swarm Optimization* -

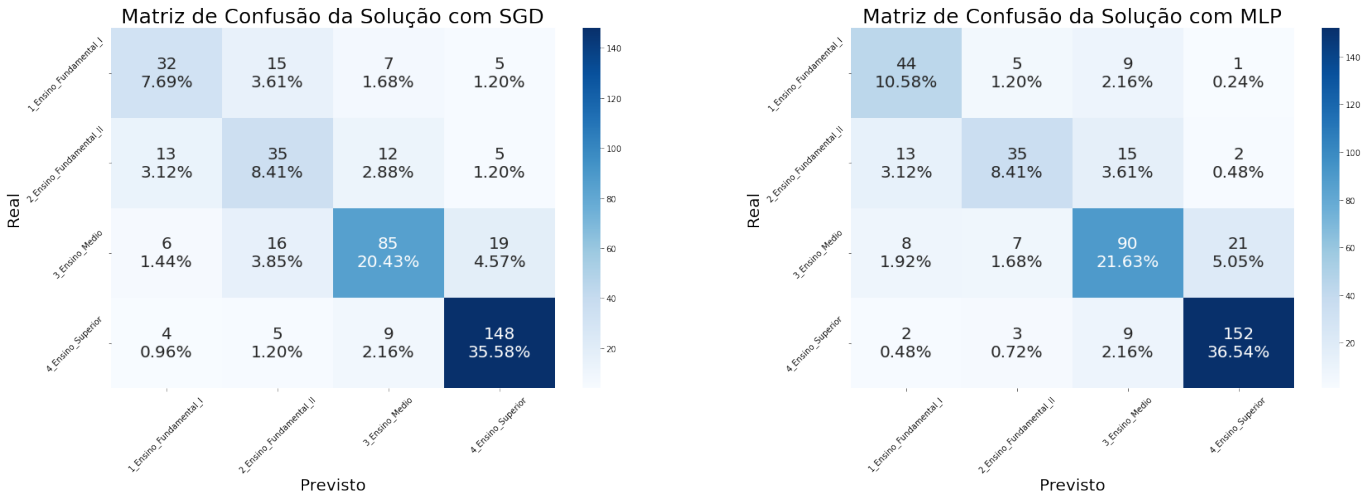


Figura 3: Matrizes de confusão do melhores algoritmos avaliados

PSO), Algoritmos Genéticos (*Genetic Algorithm* - GA) ou Otimização Bayesiana (*Bayesian Optimization* - BO).

REFERÊNCIAS

[1] Maria José Bocorny Finatto, Lucelene Lopes, and Alena Ciulla. Processamento de linguagem natural, linguística de corpus e estudos linguísticos: uma parceria bem-sucedida. *Domínios de Lingu@gem*, 9(5):41–59, ago. 2015. doi: 10.14393/DLE-v9n5a2015-3.

[2] Vinícius Augusto Alves Zanchini. *Criação de um modelo de classificação de tweets depressivos utilizando máquina de vetores de suporte.* Trabalho de conclusão de curso, Universidade Federal de Uberlândia (UFU), Patos de Minas, MG, jul 2019.

[3] Artur Silva. Classificação de texto para categorização de crimes contra a honra. In *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, pages 961–971, Porto Alegre, RS, Brasil, 2019. SBC. doi: 10.5753/eniac.2019.9349.

[4] Andrew Mccallum and Kamal Nigam. A comparison of event models for naive bayes text classification. *Work Learn Text Categ*, 752, 05 2001.

[5] Jardeson Leandro Nascimento Barbosa, João Paulo Albuquerque Vieira, Roney Lira de Sales Santos, Gilvan Veras Magalhães Junior, Mariana dos Santos Muniz, and Raimundo Santos Moura. Introdução ao processamento de linguagem natural usando python. In *Escola Regional de Informática do Piauí: Livro Anais - Artigos e Minicursos*. Escola Regional de Informática do Piauí (ERIPI), Teresina, PI, 2017.

[6] Julliano Trindade Pintas, Leandro A. F. Fernandes, and Ana Cristina Bicharra Garcia. Feature selection methods for text classification: a systematic literature review. *Artificial Intelligence Review*, 54(8):6149–6200, Dec 2021. ISSN 1573-7462. doi: 10.1007/s10462-021-09970-6. URL https://doi.org/10.1007/s10462-021-09970-6.

[7] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997. ISBN 9780071154673.

[8] Katti Faceli, Ana Carolina Lorena, João Gama, and André Carlos Ponce de Leon Ferreira de Carvalho. *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC, 2011.

[9] Rafael Ferreira-Mello, Máverick André, Anderson Pinheiro, Evandro Costa, and Cristobal Romero. Text mining in education. *WIREs Data Mining and Knowledge Discovery*, 9(6):e1332, 2019. doi: https://doi.org/10.1002/widm.1332. URL https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1332.

[10] Murilo Gazzola, Sidney Leal, Breno Pedroni, Fábio Theoto Rocha, Sabine Pompéia, and Sandra Aluísio. Text complexity of open educational resources in portuguese: Mixing written and spoken registers in a multi-task approach. *Lang. Resour. Eval.*, 56(2):621–650, jun 2022. ISSN 1574-020X. doi: 10.1007/s10579-021-09571-3. URL https://doi.org/10.1007/s10579-021-09571-3.

[11] Lele Sha, Yuheng Li, Dragan Gasevic, and Guanliang Chen. Bigger data or fairer data? augmenting BERT via active sampling for educational text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1275–1285, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.109.

[12] Bin Wang, Lmli Gao, Tao An, Mei Meng, and Tong Zhang. A method of educational news classification based on emotional dictionary. In *2018 Chinese Control And Decision Conference (CCDC)*, pages 3547–3551, 2018. doi: 10.1109/CCDC.2018.8407737.

[13] Murilo Gazzola, Sidney Evaldo Leal, and Sandra Maria Aluisio. Predição da complexidade textual de recursos educacionais abertos em português. https://github.com/gazzola/corpus_readability_nlp_portuguese, 2019. Accessed: 2021-12-05.

[14] Brasil. Lei nº 9.394, de 20 de dezembro de 1996. *Diário Oficial [da] República Federativa do Brasil*, 1996. URL http://www.planalto.gov.br/ccivil_03/leis/l9394.htm. Estabelece as diretrizes e bases da educação nacional.

[15] Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow, 2nd Edition*. Packt Publishing, 2nd edition, 2017. ISBN 1787125939.

[16] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124, 2010. doi: 10.1109/ICPR.2010.764.

[17] Guzmán Santafé, Iñaki Inza, and Jose Lozano. Dealing with the evaluation of supervised classification algorithms. *Artificial Intelligence Review*, 44, 06 2015. doi: 10.1007/s10462-015-9433-y.

[18] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2.

[19] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a.

[20] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[21] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021. doi: 10.21105/joss.03021.

[22] John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen North, Gordon Woodhull, Short Description, and Lucent Technologies. Graphviz — open source graph drawing tools. In *Lecture Notes in Computer Science*, pages 483–484. Springer-Verlag, 2001.

[23] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[24] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028, 2002.