

Utilizando IA na Identificação Problemas em Atividades em Equipe em Disciplina de Programação

Havana Diogo Alves Andrade¹, Viviane Cristina Oliveira Aureliano¹,
Patrícia Cabral de Azevedo Restelli Tedesco²

¹Instituto Federal de Pernambuco – Campus Jaboatão dos Guararapes
Jaboatão dos Guararapes – PE – Brasil

²Centro de Informática - Universidade Federal de Pernambuco
Recife – PE – Brasil

{viviane.aureliano,havana.alves}@jaboatao.ifpe.edu.br,
pcart@cin.ufpe.edu.br

Abstract. *Teaching methodologies such as Problem-Based Learning and Team-Based Learning share the formation of teams and the need for constant teacher monitoring. The submission of written reports by students is an important tool to allow this monitoring. However, reading a large number of these reports can be very time-consuming, making it difficult to identify teams that are facing some kind of problem. This article presents the use of Artificial Intelligence to analyze the reports submitted by students, informing the teacher about the problems that occur in school work teams. The results show that using these techniques can help in the faster identification of problems in teams.*

Resumo. *Metodologias de ensino como o Aprendizado Baseado em Problemas e o Aprendizado Baseado em Equipes têm em comum a formação de equipes e a necessidade de constante acompanhamento do professor. O envio de relatos escritos pelos estudantes é ferramenta importante para permitir esse acompanhamento. Porém, a leitura de um grande número destes relatos pode levar bastante tempo, dificultando a identificação das equipes que estão enfrentando algum tipo de problema. Este artigo apresenta o uso da Inteligência Artificial para analisar os relatos enviados pelos estudantes, informando ao professor sobre os problemas que ocorrem nas equipes de trabalho escolar. Os resultados mostram que utilizar estas técnicas podem ajudar na identificação mais célere de problemas nas equipes.*

1. Introdução

Estudos indicam que cursos introdutórios de programação são frequentemente considerados desafiadores por estudantes (ROBINS; ROUNTREE; ROUNTREE, 2003) e apresentam altas taxas de reprovação (BENNEDSEN; CASPERSEN, 2007; HOLANDA et al., 2021; WATSON; LI, 2014). Entre as dificuldades enfrentadas no ensino dessas disciplinas estão a falta de comunicação eficaz entre professores e alunos, bem como a ausência de feedback adequado (MEDEIROS; FALCÃO; RAMALHO, 2020).

Para reduzir a reprovação média de 68% registrada ao longo de sete semestres em uma disciplina introdutória de programação, abordagens como Aprendizado Baseado em Problemas (ABP) (DA CONCEIÇÃO; GUEDES, 2021) e Aprendizado Baseado em Equipes (ABE) (ALVES; JÚNIOR, 2018) substituíram aulas predominantemente expositivas. Pesquisas mostram que atividades colaborativas

promovem melhor desempenho em tarefas complexas (ZAMBRANO et al., 2019) e que a resolução colaborativa de problemas pode melhorar os resultados cognitivos e afetivos dos estudantes (LAI; WONG, 2022).

Tanto o ABP quanto o ABE enfatizam a autonomia do estudante e o trabalho em grupo. No entanto, desafios como conflitos internos, dificuldades de comunicação e falta de coesão podem comprometer a eficácia dessas metodologias. A coesão entre os membros dos grupos é essencial para o sucesso das equipes, promovendo confiança, troca de conhecimento e desenvolvimento de ideias (ZAMECNIK et al., 2021). Para serem aplicadas, a ABP e a ABE exigem acompanhamento constante por parte dos professores, contudo monitorar vários grupos em turmas numerosas é desafiador. No mundo ideal, os estudantes deveriam fornecer *feedback* de maneira proativa e frequente sobre suas atividades e interações em equipe. No entanto, na prática, problemas só são relatados nos estágios finais da atividade ou da disciplina, quando intervenções eficazes são limitadas.

Uma solução potencial seria uma plataforma que permitisse aos alunos fornecer relatos frequentes e confidenciais, acessíveis apenas ao professor. Ainda assim, somente disponibilizar tal ferramenta não garante o envio de *feedbacks* significativos. Além disso, o aumento da carga de trabalho para a leitura e análise de relatos pode tornar a tarefa de difícil realização, dificultando a identificação de problemas rapidamente.

Técnicas de Processamento de Linguagem Natural (PLN), como Análise de Sentimentos (AS), podem ser úteis para automatizar a avaliação dos *feedbacks* dos estudantes pelos professores, identificando os sentimentos expressos pelos estudantes e facilitando o trabalho docente (ALLEN, 1995; JURAFSKY; MARTIN, 2021).

Dentro desse contexto, este trabalho propõe uma abordagem que lança mão de técnicas de PLN para automatizar a AS em relatos de estudantes matriculados em uma disciplina introdutória de programação, fornecidos após a realização de atividades em grupos. A abordagem possibilita o auxílio aos professores na identificação precoce de sentimentos negativos expressos pelos estudantes durante atividades em grupo, permitindo intervenções rápidas e direcionadas, sendo essa a principal contribuição deste trabalho.

2. Análise de sentimentos

Análise de Sentimentos (AS) é a tarefa de extrair, utilizando técnicas de PLN, orientação sobre a emoção ou a opinião expressa em um texto em relação a determinado objeto (JURAFSKY; MARTIN, 2021). É amplamente utilizada para mineração de opinião ou emoção. A mineração de emoções é mais complexa devido às seis emoções básicas propostas por (EKMAN, 1999): surpresa, felicidade, raiva, medo, nojo e tristeza. Já a mineração de opinião foca na polaridade textual – positiva, negativa ou neutra (YADOLLAHI; SHAHRAKI; ZAIANE, 2017). Essas classificações têm aplicações relevantes em áreas como vendas, política e mercado financeiro (ABBASI-MOUD; VAHDAT-NEJAD; SADRI, 2021; CAROSIA; COELHO; DA SILVA, 2021; RAHARDJA; HARIGUNA; BAIHAQI, 2019) e educação (DOLIANITI et al., 2019; KASTRATI et al., 2021). O alvo da AS pode ser um documento inteiro

(*document-level*), uma frase (*sentence-level*) ou mesmo palavras individuais (*word-level*) (BEHDENNA; BARIGOU; BELALEM, 2018; FARRA et al., 2010).

Existem três classes principais de abordagens de AS: (i) abordagens baseadas em léxico ou em conhecimento; (ii) algoritmos de aprendizado de máquina (AM); e (iii) métodos híbridos (CAMBRIA, 2016; CHAI; LI, 2019; MATHEW; BINDU, 2020). As abordagens baseadas em léxico utilizam dicionários que atribuem valores quantitativos (e.g., -1/0/1) ou qualitativos (e.g., positivo/neutro/negativo) às palavras para indicar o grau de polaridade sentimental. As abordagens que utilizam AM normalmente lançam mão de algoritmos de aprendizagem supervisionados. Por fim, as abordagens híbridas utilizam os dois

3. Trabalhos relacionados

A AS tem se mostrado uma ferramenta valiosa para obter *insights* sobre emoções e opiniões de estudantes em contextos educacionais. Suas aplicações incluem a avaliação de cursos, suporte à tomada de decisão institucional, personalização de experiências educacionais e fornecimento de feedback aos estudantes (DOLIANITI et al., 2019). Pesquisas sobre AS em feedback estudantil frequentemente investigam aspectos como opiniões sobre professores, cursos e instituições de ensino (KASTRATI et al., 2021).

Dake e Gyimah (2023) utilizaram AS para avaliar 232 *feedbacks* de uma disciplina de Aplicações e Sistemas de Computadores, analisando respostas a perguntas abertas com quatro algoritmos de aprendizado supervisionado: *Random Forest*, *J48 Decision Tree*, *Naive Bayes* e *Support Vector Machine* (SVM). O SVM apresentou a maior acurácia, alcançando 63,79%. Os textos analisados estavam em língua inglesa.

Pfitscher et al. (2023) empregaram AS para compreender sentimentos e reduzir taxas de evasão em cursos de programação no ensino superior brasileiro. Foi utilizada a arquitetura de rede neural *Long Short Term Memory* (LSTM), o qual foi treinada utilizando duas bases de textos em português: a de resenhas de filmes do site IMDb (Gonçalves, 2018) e o SenticNet (Cambria e Hussain, 2022), que consiste de palavras e termos individuais. Após coletar 158 textos de *feedback*, em língua portuguesa, de duas instituições, aplicaram análises psicopedagógicas manuais e automatizadas, observando uma distribuição de 20-30% de sentimentos negativos e alcançando 68% de acurácia com a análise automatizada.

Oliveira et al. (2023) examinaram sentimentos expressos por professores em mensagens de fóruns de uma disciplina sobre tecnologias educacionais. Os autores realizaram análise léxica sobre 1066 mensagens escritas em português, utilizando a biblioteca *Syuzhet*¹. Não foi indicada taxa de acurácia do experimento.

Lundqvist et al. (2020) exploraram o uso de AS automatizada para avaliar o *feedback* de estudantes em uma disciplina de programação ofertada em um MOOC. Os autores compararam postagens online com *feedbacks* fornecidos (em língua inglesa) e observaram que estudantes iniciantes demonstraram sentimentos mais positivos em relação à disciplina do que estudantes mais experientes. Foi utilizada a abordagem léxica por não existirem bases de dados de textos relativos a MOOCs disponíveis.

¹ <https://cran.r-project.org/web/packages/syuzhet/index.html>

Watkins et al. (2020) propuseram o SENSE (*Student pErformance quaNtifier using SEntiment analysis*), que usa PLN e o pacote VADER para quantificar feedbacks de professores a alunos e seus pais. A análise foi realizada em textos em inglês, utilizando abordagem léxica.

Um dos grandes problemas para se realizar a AS sobre textos na área educacional é a falta de base de dados pré-existentes específicos para este contexto, que possam auxiliar na fase de treinamento de algoritmos de AM, tanto para língua portuguesa quanto inglesa - como constatado por Lundqvist et al. (2020). Assim, alguns pesquisadores optam por utilizar abordagens léxicas, como Oliveira et al. (2023), Lundqvist et al. (2020) e Watkins et al. (2020), enquanto outros utilizam textos coletados para treinar algoritmos de AM. Porém, a quantidade de textos pode ter relação direta com o valor de acurácia dos algoritmos. Dake e Gyimah (2023) e Pfitscher et al. (2023) obtiveram menos de 70% de acurácia ao treinar algoritmos com apenas algumas centenas de exemplares.

Em nosso trabalho, realizamos a classificação a nível de sentença, pois a divisão dos textos em sentenças mais que dobrou a quantidade de exemplares. Além disso, métodos de superamostragem foram utilizados para que fosse possível realizar o treinamento de algoritmos de AM de forma mais eficaz. Estes procedimentos ajudaram a obter acurácia dos algoritmos testados de 74 a 99%. Todo o processo é explanado na próxima seção.

4. Metodologia

A coleta de relatos dos estudantes iniciou-se em 2017 e seguiu em 2018, de maneira voluntária por parte do estudante, no entanto, poucos textos foram enviados de forma proativa neste período. Esta falta de adesão à escrita de *feedbacks* sobre suas atividades foi percebida também em Pfitscher et al. (2023), onde apenas 27% dos estudantes responderam ao questionário da referida pesquisa. Assim, no ano de 2019, o envio dos relatos tornou-se uma atividade obrigatória e incorporada à avaliação da disciplina, o que aumentou a demanda de leitura dos relatos por parte da docente, com o objetivo de identificar possíveis problemas, de forma o mais célere possível. Os relatos dos estudantes foram coletados por meio da plataforma Moodle², durante e após as atividades em grupo realizadas na disciplina, utilizando o plugin Diário de Bordo (Journal³) disponível na plataforma.

Ao final de 3 anos de coleta, 651 relatos foram obtidos no total, quantidade que pode ser considerada baixa para o treinamento de algoritmos de AM. Assim, os textos foram divididos em sentenças. Após a remoção de duplicatas, estes relatos resultaram em 1.363 sentenças, que foram o alvo da AS realizada neste trabalho. Estas sentenças foram classificadas manualmente (*gold labels*) entre apresentar um sentimento positivo ou negativo, resultando em 1.206 sentenças com sentimento positivo contra 157 sentenças com sentimentos negativos.

² <https://moodle.org/>

³ https://moodle.org/plugins/mod_journal

Estas sentenças passaram então por um pré-processamento composto por 5 etapas: (i) correções ortográficas e gramaticais; (ii) tokenização e POS-tagging; (iii) remoção de *stop-words*; (iv) lematização dos tokens de palavras; e (v) limpeza básica final. As sentenças originais apresentavam erros ortográficos e gramaticais, e por isso, elas foram corrigidas. Em seguida, ocorreu a tokenização, na qual foram identificados os tokens válidos das sentenças e, logo após, a POS-tagging, na qual os tokens identificados foram classificados de acordo com a sua classe gramatical. Na etapa de remoção de *stop-words*, foram retirados os tokens que não acrescentavam valor a AS das sentenças. Em seguida, na etapa de lematização dos tokens de palavras, as palavras foram reduzidas às suas formas base para padronizar variações do mesmo termo, fazendo com que palavras diferentes, mas com mesma raiz seguissem com a mesma grafia (BALAKRISHNAN; LLOYD-YEMOH, 2014). Na etapa de limpeza básica, foram excluídas as sentenças que resultaram em uma única palavra ou um único caracter. Assim, ao final do pré-processamento, restaram 1.295 sentenças, distribuídas entre 1.138 sentenças classificadas como opiniões positivas e 157 como opiniões negativas.

Para a classificação de sentimentos, foram testados vários algoritmos supervisionados de AM, incluindo Regressão Logística, *Gradient Boosting*, *KNearest Neighbor* (KNN), Árvores de Decisão e *Support Vector Machines* (SVM). Os algoritmos de AM escolhidos, e suas variações, são apresentados na lista abaixo:

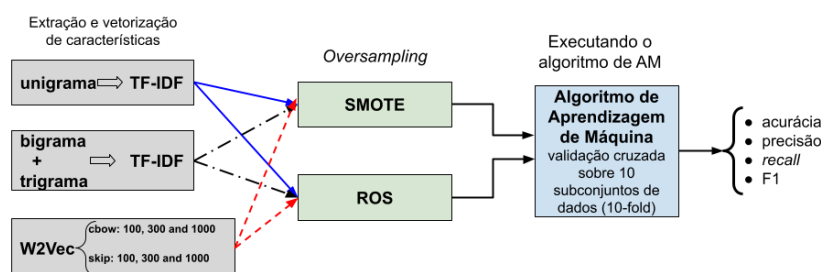
- Regressão logística, utilizando os solucionadores *large linear*, *newton*, *limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS)* e *stochastic average gradient (SAG)*;
- *Gradient Boosting* com o número de estimadores 10, 50, 100, 500 e 1000;
- *K-Nearest Neighbor (KNN)* com 1, 2, 3, 5 e 7 vizinhos foram testados e comparados;
- Árvores de Decisão utilizando os critérios *gini*, *entropy* e *log loss*, combinados com 3, 5, 7, 9, 11 e 13 níveis de profundidade máxima;
- *Support Vector Machine (SVM)* utilizando os kernels *linear*, *polynomial* e *RBF*.

Assim, para cada um destes algoritmos, foi aplicado o pipeline descrito na Figura 1 que consiste nos seguintes passos:

1. O conjunto de dados foi vetorizado com *Term Frequency-Inverse Document Frequency* (TF-IDF) e com *Word2Vec*, seguindo as seguintes configurações:
 - a. **N-gram e TF-IDF**: sendo uma vez por unigramas e uma vez por combinações de bigramas e trigramas (bi-trigramas). O vetor gerado a partir dos unigramas resultou em 2.638 colunas. A aplicação de bi-trigramas resultou em um vetor de 24.251 colunas. Ambas as matrizes possuem 1.295 linhas; e
 - b. **Word2Vec** com *embeddings* de palavras pré-treinadas com base em CBOW e SKIPGRAM (MIKOLOV, 2013), ambos gerados com 100, 300 e 1000 dimensões. O resultado desta etapa foram oito vetores de características diferentes.

2. Devido ao desbalanceamento entre a quantidade de sentenças positivas e negativas, cada vetor resultante das técnicas anteriores foi então superamostrado com base no algoritmo de geração aleatória *Random Oversampling of Minority Classes*- ROS e de geração sintética *Synthetic Minority Oversampling Technique* -SMOTE (CHAWLA et al., 2002). Ambos os métodos criaram mais 981 linhas para aumentar a classe minoritária, resultando em 2.276 amostras.
3. Cada um desses conjuntos de dados superamostrados foi usado como entrada para os modelos de classificação escolhidos, para serem treinados e testados em um algoritmo de validação cruzada sobre 10 subconjuntos de dados (*10-fold cross-validation*).

Figura 1. Pipeline aplicado ao conjunto de dados.



As métricas de avaliação incluíram acurácia, precisão, *recall* (principal métrica considerada do estudo) e F1-Score. A implementação foi realizada em Python, utilizando bibliotecas de PLN como SpaCy, NLTK e *Language Tool*⁴ (MIŁKOWSKI, 2010) para o pré-processamento, scikit-learn para treinamento e validação dos modelos.

5. Resultados

Com base no conjunto de dados resultante, cinco algoritmos de AM foram testados em múltiplas configurações, e seus desempenhos foram avaliados usando as métricas de acurácia, precisão, recall e F1-score. O foco principal foi maximizar o *recall* para a detecção de sentimentos negativos, dada a importância de identificar potenciais problemas nas equipes de estudantes. No **Quadro 1** é apresentado um resumo das métricas encontradas para todas as configurações testadas.

O melhor valor de *recall* - 0,82571 - foi encontrado após a execução da Regressão Logística com solver linear no conjunto de dados vetorizado por unigramas e TF-IDF, e posteriormente super amostrado pelo método ROS. As demais métricas para esta execução também apresentaram bons valores: acurácia = 0,97033; precisão = 0,78879; e F1 = 0,80485.

O valor de F1 próximo de 1 mostra que há um certo equilíbrio entre a precisão e o recall. Para analisar a qualidade deste modelo, traçamos uma curva *Receiver Operator*

⁴ Language tool. Disponível em: <<https://pypi.org/project/language-tool-python/>>. Acesso em: 13 dez. 2024.

Characteristic (ROC), ideal para indicar o quanto o modelo é capaz de realizar a distinção entre as classes de um problema de classificação (FAWCETT, 2006) e para modelos com classes desbalanceadas. A **Figura 2** mostra a curva ROC do modelo que encontrou o melhor valor de *recall*. O valor ROC-AUC calculado é de 0,97925, o que prova que o modelo é adequado para classificar um conjunto de dados como o utilizado neste trabalho.

5.1. Aplicação de uso do modelo

O modelo de AM que teve a melhor performance e também o vetor de características gerado foram salvos em arquivos do tipo *pickle* (onde objetos Python são convertidos em um fluxo de bytes). Estes arquivos permitem a perpetuação do modelo e do vetor, os quais serão usados para classificar novos textos. Como forma de automatizar a análise de novos relatos, uma plataforma web foi criada, onde os estudantes poderão inserir seus relatos e a um determinado intervalo de tempo (a cada 10 minutos), um script do servidor realiza pesquisa no banco de dados em busca de novos relatos que ainda não foram classificados. Cada texto de *feedback* passa pelas mesmas etapas de pré-processamento explicadas na seção de Metodologia e então é classificado pelo modelo de AM que foi salvo.

Os resultados das análises dos relatos são, então, enviados para o e-mail do professor. Se alguma sentença do texto de *feedback* for classificada como contendo um problema (sentimento negativo), o texto será escrito em fonte vermelha. Caso nenhum problema seja identificado, o texto é enviado em fonte verde. O texto na **Figura 3** expressa um sentimento positivo. A **Figura 4** exibe um relato em que a primeira sentença contém um sentimento negativo, enquanto que a segunda sentença expressa uma opinião positiva. Neste caso, foi realizada intervenção na equipe por parte da professora para saber o porquê de um integrante ter saído do grupo, e se seria viável algum tipo de reorganização.

5.2. Discussão

Os resultados obtidos demonstram o potencial da AS como uma ferramenta para monitorar o progresso de equipes em disciplinas introdutórias de programação. A aplicação de técnicas de PLN e AM permitiu identificar sentimentos negativos nos relatos dos estudantes, o que tornaria possível intervenções mais ágeis e eficazes por parte do professor. O modelo que obteve o melhor desempenho – Regressão Logística com vetorização TF-IDF e superamostragem ROS – alcançou um *recall* de 82,57%, o que reflete a capacidade do sistema em detectar potenciais problemas nas equipes. O equilíbrio entre precisão e *recall*, evidenciado pelo F1-Score de 0,80485, reforça a confiabilidade do modelo.

A utilização de vetores TF-IDF e *embeddings* de palavras mostrou-se eficaz na extração de características relevantes dos dados textuais, enquanto o uso de técnicas de superamostragem, como SMOTE e ROS, mitigou o desbalanceamento entre as classes de sentimento. Apesar desses avanços, algumas limitações devem ser consideradas. O pequeno volume de dados e a necessidade de rotulação manual restringiram a escalabilidade da abordagem. Adicionalmente, o foco exclusivo em textos escritos em

português, embora relevante para o contexto do estudo, dificulta a comparação direta com outros trabalhos na literatura, predominantemente conduzidos em inglês.

Para superar essas restrições, futuros estudos podem explorar o uso de LLMs, como o ChatGPT⁵, que têm demonstrado resultados promissores na tarefa de AS (ZHANG et al., 2023). Além disso, a expansão do conjunto de dados por meio de coletas adicionais ou colaboração com outras instituições pode aumentar a robustez e a generalização dos resultados.

Outro ponto importante a ser levantado é a adesão dos estudantes à escrita dos diários reflexivos. Observou-se que a obrigatoriedade do envio, aliada à incorporação na avaliação, aumentou a participação. Contudo, estratégias complementares, como gamificação ou *feedback* imediato automatizado, podem ser investigadas para motivar os alunos a contribuir de forma mais espontânea e consistente.

Figura 2. Curva ROC do modelo de Regressão Logística.

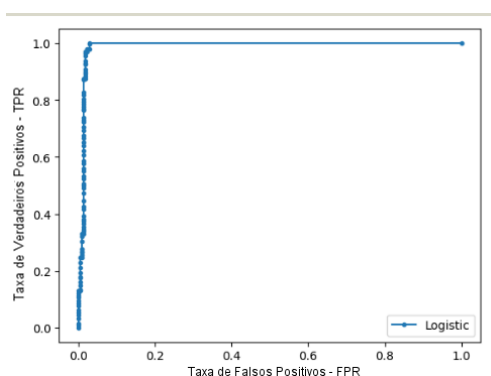


Figura 3. E-mail enviado a docente contendo sentença com sentimento classificado como positivo

Meu grupo está estudando bastante e separamos um dia na semana Para estudarmos juntos em ligação

Figura 4. E-mail enviado a docente contendo sentença com sentimento classificados como negativo e outra sentença classificada como negativa

o grupo foi criado com 4 integrantes, porém 1 deles sumiu, simplesmente não sabemos se ele desistiu, então seguimos com os 3 integrantes. Após a primeira semana de aula os integrantes mantiveram mais contato e vêm trocando experiências a partir dos exercícios passados em aula.

Quadro 1. Métricas encontradas para todas as configurações testadas.

⁵ ChatGPT. Disponível em: <<https://chatgpt.com/>>. Acesso em: 17 dez. 24.

Model	Vectorization Method	Oversampling Method	Accuracy	Precision	Recall	F1	Observation
Logistic Regression	Unigram TFIDF	ROS	0.97033	0.78879	0.82571	0.80485	Solver = linear
	W2Vec	SMOTE	0.98791	0.84900	0.73214	0.7862	Solver = linear
Gradient Boosting	Unigram TFIDF	ROS	0.95604	0.79937	0.66666	0.72701	estimators=1000
	W2Vec	ROS	0.98571	0.83275	0.66666	0.74051	estimators=500, CBOW 300 dimensions
Decision Tree	Unigram TFIDF	ROS	0.87692	0.77728	0.64732	0.70064	max. depth = 13 criterion = gini
	W2Vec	ROS	0.89396	0.80052	0.65951	0.72321	max. depth = 13 criterion = gini dimensions
SVM Linear	Unigram TFIDF	ROS	0.99670	0.83041	0.66666	0.73958	kernel = linear
	W2Vec	ROS	0.90495	0.80223	0.66523	0.72734	kernel = linear CBOW 100 dimensions
SVM Poly	Bi-Trigram TFIDF	ROS	0.99780	0.83041	0.66666	0.73958	kernel = polynomial
	W2Vec	SMOTE	0.74725	0.72991	0.65379	0.68976	kernel = polynomial CBOW 100 dimensions
SVM RBF	Bi-Trigram TFIDF	ROS	0.99725	0.83041	0.66666	0.73958	kernel = polynomial
	W2Vec	ROS	0.93077	0.81678	0.643776	0.72003	kernel = rbf CBOW 300 dimensions
KNearest Neighbor	Bi-Trigram TFIDF	ROS	0.93681	0.79574	0.66666	0.72552	neighbors=1
	W2Vec	ROS	0.96868	0.82459	0.66666	0.73726	neighbors = 1, CBOW 300 dimensions

Por fim, os resultados indicam que a AS automatizada é uma abordagem promissora não apenas para identificar problemas em tempo hábil, mas também para fomentar reflexões nos estudantes sobre suas interações em equipe. Essa ferramenta

pode ser adaptada para outras disciplinas ou contextos educacionais, ampliando seu impacto potencial no monitoramento e na melhoria do ensino colaborativo.

6. Considerações finais

Neste trabalho, apresentamos uma abordagem que utilizou a Análise de Sentimentos (AS) para classificar os relatos de estudantes do ensino médio técnico durante atividades em equipe em uma disciplina introdutória de programação. O conjunto de dados consistiu em 1.363 sentenças, das quais 1.206 foram classificadas como positivas e 157 como negativas. Para a AS, empregaram-se modelos supervisionados de Aprendizado de Máquina (AM), incluindo Regressão Logística, Gradient Boosting, KNN, Árvores de Decisão e Support Vector Machines (SVM). A vetorização foi realizada com TF-IDF e Word2Vec, e as técnicas de superamostragem ROS e SMOTE foram aplicadas para lidar com o desbalanceamento de classes.

O modelo com melhor desempenho, medido pelo recall (0,82571), foi a Regressão Logística com *solver* linear, utilizando unigramas vetorizados por TF-IDF e superamostragem com ROS. Esse modelo foi integrado a uma plataforma web que automatiza a identificação de sentimentos negativos nos relatos, enviando os resultados por e-mail ao professor. Sentenças com sentimentos negativos são destacadas em fonte vermelha, enquanto textos sem problemas são exibidos em fonte verde, permitindo intervenções rápidas e eficazes.

A principal dificuldade identificada foi a escassez de conjuntos de dados textuais em português para treinar modelos de AM, o que evidenciou uma lacuna na literatura acadêmica que merece exploração futura. Observou-se também que os estudantes, em sua maioria, não fornecem feedbacks espontaneamente, sendo necessária a intervenção constante do professor para incentivar a utilização da ferramenta. A obrigatoriedade do envio dos relatos, incorporada à avaliação da disciplina, mostrou-se a estratégia mais eficiente para estimular a participação, contribuindo para a identificação precoce de problemas.

Atualmente, a abordagem está sendo aplicada em turmas de atividades em grupo, com o objetivo de avaliar, ao final do semestre, a eficácia da AS na resolução de conflitos e na mitigação de obstáculos que impactam o desempenho escolar. Como contribuição adicional, pretende-se disponibilizar os relatos tratados em plataformas como o Kaggle⁶, garantindo a remoção de quaisquer dados sensíveis que comprometam a privacidade dos alunos.

⁶ <https://www.kaggle.com/>

Referências

- ABBASI-MOUD, Z.; VAHDAT-NEJAD, H.; SADRI, J. Tourism recommendation system based on semantic clustering and sentiment analysis. **Expert Systems with Applications**, v. 167, p. 114324, 2021.
- ALLEN, J. **Natural language understanding**. [s.l.] Benjamin-Cummings Publishing Co., Inc., 1995.
- ALTRABSHEH, N.; COCEA, M.; FALLAHKHAIR, S. **Sentiment analysis: towards a tool for analysing real-time students feedback**. 2014 IEEE 26th international conference on tools with artificial intelligence. **Anais...IEEE**, 2014.
- ALVES, H.; JÚNIOR, F. N. Problem Based Learning e Scratch: aliados no ensino de Logica de Programação. 2018.
- AUNG, K. Z.; MYO, N. N. **Sentiment analysis of students' comment using lexicon based approach**. 2017 IEEE/ACIS 16th international conference on computer and information science (ICIS). **Anais...IEEE**, 2017.
- BALAKRISHNAN, V.; LLOYD-YEMOH, E. Stemming and lemmatization: A comparison of retrieval performances. 2014.
- BEHDENNA, S.; BARIGOU, F.; BELALEM, G. Document Level Sentiment Analysis: A survey. **EAI Endorsed Transactions on Context-aware Systems and Applications**, v. 4, n. 13, p. e2, mar. 2018.
- BENNEDSEN, J.; CASPERSEN, M. E. Failure rates in introductory programming. **AcM SIGCSE Bulletin**, v. 39, n. 2, p. 32–36, 2007.
- CAMBRIA, E.; HUSSAIN, A. **SenticNet 7: A Commonsense-based Neurosymbolic AI Framework for Explainable Sentiment Analysis**, Proceedings of LREC, p. 3829-399, 2022.
- CAMBRIA, E. Affective computing and sentiment analysis. **IEEE Intelligent Systems**, n. 2, 2016.
- CAROSIA, A. E. O. E.; COELHO, G. P.; DA SILVA, A. E. A. Investment strategies applied to the Brazilian stock market: a methodology based on sentiment analysis with deep learning. **Expert Systems with Applications**, v. 184, p. 115470, 2021.
- CHAI, J.; LI, A. **Deep Learning in Natural Language Processing: A State-of-the-Art Survey**. 2019 International Conference on Machine Learning and Cybernetics (ICMLC). **Anais...2019**.
- CHAUHAN, G.; AGRAWAL, P.; MEENA, Y. Aspect-Based Sentiment Analysis of Students' Feedback to Improve Teaching–Learning Process: Proceedings of ICTIS 2018, Volume 2. Em: [s.l.: s.n.]. p. 259–266.
- CHAWLA, N. V. et al. SMOTE: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, v. 16, p. 321–357, 2002.
- DA CONCEIÇÃO, C. M.; GUEDES, A. M. A. Aprendizagem Baseada em Problemas aplicada à Programação de Computadores: Um Mapeamento Sistemático. **RENOTE**, v. 18, n. 2, p. 570–580, jan. 2021.
- DAKE, D. K.; GYIMAH, E. Using sentiment analysis to evaluate qualitative students' responses. **Education and Information Technologies**, v. 28, n. 4, p. 4629–4647, 1 abr. 2023.
- DOLIANITI, F. S. et al. **Sentiment Analysis Techniques and Applications in Education: A Survey**. (M. Tsitouridou, J. A. Diniz, T. A. Mikropoulos, Eds.)Technology and Innovation in Learning, Teaching and Education. **Anais...Cham: Springer International Publishing**, 2019.
- EKMAN, P. Basic emotions. **Handbook of cognition and emotion**, v. 98, n. 45–60, p. 16, 1999.
- FARRA, N. et al. **Sentence-Level and Document-Level Sentiment Mining for Arabic Texts**. 2010 IEEE International Conference on Data Mining Workshops. **Anais...2010**.
- FAWCETT, T. An introduction to ROC analysis. **Pattern recognition letters**, v. 27, n. 8, p. 861–874, 2006.
- GONÇALVES, L. **Imdb pt-br**. url=<https://www.kaggle.com/datasets/luisfredgs/imdbptbr>, 2018.
- HOLANDA, M. et al. **Educational Initiatives to Increase Diversity in CS1 Courses: A Literature Mapping of U.S. efforts**. **Proceedings - Frontiers in Education Conference, FIE**, 2021. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85123835606&doi=10.1109%2fFIE49875.2021.9637445&partnerID=40&md5=5e14d92aec3a4bb978b1659be8e27c49>>
- JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing**. [s.l.: s.n.].
- KASTRATI, Z. et al. Sentiment Analysis of Students' Feedback with NLP and Deep Learning: A Systematic Mapping Study. **Applied Sciences**, v. 11, n. 9, 2021.
- LAI, X.; WONG, G. K. -W. Collaborative versus individual problem solving in computational thinking through programming: A meta-analysis. **British Journal of Educational Technology**, v. 53, n. 1, p. 150–170, 2022.

LI, J. et al. A Survey on Deep Learning for Named Entity Recognition. **IEEE Transactions on Knowledge and Data Engineering**, v. 34, n. 1, p. 50–70, 2022.

LIDDY, E. D. Natural language processing. 2001.

LUNDQVIST, K.; LIYANAGUNAWARDENA, T.; STARKEY, L. Evaluation of Student Feedback Within a MOOC Using Sentiment Analysis and Target Groups. **The International Review of Research in Open and Distributed Learning**, v. 21, n. 3, p. 140–156, maio 2020.

MATHEW, L.; BINDU, V. R. **A Review of Natural Language Processing Techniques for Sentiment Analysis using Pre-trained Models**. 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC). **Anais...**2020.

MEDEIROS, R.; FALCÃO, T.; RAMALHO, G. **Ensino e Aprendizagem de Introdução à Programação no Ensino Superior Brasileiro: Revisão Sistemática da Literatura**. Anais do XXVIII Workshop sobre Educação em Computação. **Anais...**Porto Alegre, RS, Brasil: SBC, 2020. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/11155>>

MIKOLOV, T. **Efficient Estimation of Word Representations in Vector Space**. ArXiv.org, 7 Sept. 2013, arxiv.org/abs/1301.3781.

MIŁKOWSKI, M. **Developing an open-source, rule-based proofreading tool**. Software: Practice and Experience, v. 40, n. 7, p. 543–566, 2010.

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. **Natural language processing: an introduction**. Journal of the American Medical Informatics Association, v. 18, n. 5, p. 544–551, set. 2011.

NAVIGLI, R. Word Sense Disambiguation: A Survey. **ACM Comput. Surv.**, v. 41, n. 2, fev. 2009.

OLIVEIRA, E. et al. **Análise de Sentimentos e Emoções em Fóruns Educacionais: Um Estudo de Caso em um Curso de Formação Docente durante a Pandemia**. Anais do XXIX Workshop de Informática na Escola. **Anais...**Porto Alegre, RS, Brasil: SBC, 2023. Disponível em: <<https://sol.sbc.org.br/index.php/wie/article/view/26367>>

PFITSCHER, R. et al. **Análise de sentimentos em turmas de programação com vistas ao apoio à permanência estudantil**. Anais do XXXIV Simpósio Brasileiro de Informática na Educação. **Anais...**Porto Alegre, RS, Brasil: SBC, 2023.

RAHARDJA, U.; HARIGUNA, T.; BAIHAQI, W. M. **Opinion Mining on E-Commerce Data Using Sentiment Analysis and K-Medoid Clustering**. 2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media). **Anais...**2019.

ROBINS, A.; ROUNTREE, J.; ROUNTREE, N. Learning and teaching programming: A review and discussion. **Computer science education**, v. 13, n. 2, p. 137–172, 2003.

ROGERS, P.; LEA, M. Cohesion in online groups. **WIT Transactions on Information and Communication Technologies**, v. 31, 2004.

SALAS, E. et al. Measuring team cohesion: Observations from the science. **Human factors**, v. 57, n. 3, p. 365–374, 2015.

SANTOS, F.; SILVEIRA, I. F.; LECHUGO, C. **Mineração da percepção do aluno: Mineração de Dados Educacionais na geração de indicadores para a avaliação de práticas pedagógicas docentes**. Anais dos Workshops do Congresso Brasileiro de Informática na Educação. **Anais...**2016.

SPARCK-JONES, K. S.; WILLETT, P.; OTHERS. **Readings in information retrieval**. [s.l.] Morgan Kaufmann, 1997.

WANG, Y. A Comprehensive Survey of Grammatical Error Correction. **ACM Trans. Intell. Syst. Technol.**, v. 12, n. 5, dez. 2021.

WATKINS, J.; FABIELLI, M.; MAHMUD, M. **SENSE: a Student Performance Quantifier using Sentiment Analysis**. 2020 International Joint Conference on Neural Networks (IJCNN). **Anais...**2020.

WATSON, C.; LI, F. W. B. **Failure rates in introductory programming revisited**. Proceedings of the 2014 conference on Innovation & technology in computer science education. **Anais...**2014.

WILLIAMS, E. A.; DURAY, R.; REDDY, V. Teamwork orientation, group cohesiveness, and student learning: A study of the use of teams in online distance education. **Journal of Management Education**, v. 30, n. 4, p. 592–616, 2006.

YADOLLAHI, A.; SHAHRAKI, A. G.; ZAIANE, O. R. Current state of text sentiment analysis from opinion to emotion mining. **ACM Computing Surveys (CSUR)**, v. 50, n. 2, p. 1–33, 2017.

ZAMBRANO, J. et al. Effects of prior knowledge on collaborative and individual learning. **Learning and Instruction**, v. 63, p. 101214, 2019.

ZAMECNIK, A. et al. The cohesion of small groups in technology-mediated learning environments: A systematic literature review. **Educational Research Review**, p. 100427, 2021.

ZHANG, W. et al. **Sentiment Analysis in the Era of Large Language Models: A Reality Check**. arXiv, , 24 maio 2023. Disponível em: <<http://arxiv.org/abs/2305.15005>>. Acesso em: 17 dez. 2024