

Tutor Inteligente: Machine Learning Aplicada ao Processo de Ensino e Aprendizagem da Matemática no Ensino Básico

Paulo Ricardo Mesquita Rosa Santos
Universidade Veiga de Almeida – UVA
Rio de Janeiro, Brasil
0009-0002-6003-5371

Vinícius Marques da Silva Ferreira
Instituto Oswaldo Cruz – FioCruz
Rio de Janeiro - RJ
0000-0003-3664-3510

Luiz Anastácio Alves
Instituto Oswaldo Cruz – FioCruz
Rio de Janeiro, Brasil
0000-0002-0785-7272

Alfredo Nazareno Pereira Boente
Universidade Federal do Rio de Janeiro – UFRJ
Rio de Janeiro, Brasil
0000-0002-2718-4917

João Jorge Sarto Neto
Universidade Iguazu – UNIG
Rio de Janeiro, Brasil
0009-0004-1938-7003

Jean Marcos Rebello Marinho
Universidade Veiga de Almeida – UVA
Rio de Janeiro, Brasil
0009-0005-8154-9023

Abstract—This article presents the development of Tutor Inteligente, a system based on machine learning, created to personalize the teaching-learning process of mathematics in basic education. The methodology adopted involves machine learning techniques for analyzing and adapting content to students' individual needs. The development of the system follows an incremental approach with agile methodologies (SCRUM), including continuous testing (TDD) and integration with real-time assessment tools. Partial results show that the Intelligent Tutor provides immediate feedback and adaptive support, promoting a more effective and interactive learning experience for students with specific difficulties in mathematics. The conclusion points to the system as a complementary tool to teaching, capable of enriching the educational process and increasing knowledge retention.

Keywords—Machine Learning; Intelligent Tutoring; Adaptive Education; Mathematics in Basic Education; Education 4.0.

Resumo—Este artigo apresenta o desenvolvimento do Tutor Inteligente, um sistema baseado em machine learning, criado para personalizar o processo de ensino-aprendizagem da matemática no ensino básico. A metodologia adotada envolve técnicas de aprendizado de máquina para análise e adaptação do conteúdo às necessidades individuais dos alunos. O desenvolvimento do sistema segue uma abordagem incremental com metodologias ágeis (SCRUM), incluindo testes contínuos (TDD) e integração com ferramentas de avaliação em tempo real. Os resultados parciais mostram que o Tutor Inteligente fornece feedback imediato e suporte adaptativo, promovendo uma experiência de aprendizado mais eficaz e interativa para alunos com dificuldades específicas na matemática. A conclusão aponta o sistema como uma ferramenta complementar ao ensino, capaz de enriquecer o processo educacional e aumentar a retenção de conhecimento.

Palavras-chave—Aprendizado de Máquina; Tutoria Inteligente; Educação Adaptativa; Matemática no Ensino Básico; Educação 4.0.

I. INTRODUÇÃO

No contexto educacional contemporâneo, a incorporação de tecnologias digitais no processo de ensino-aprendizagem tem se mostrado uma estratégia eficaz para potencializar o aprendizado dos estudantes, pois dentre essas tecnologias, a aplicação de sistemas baseados em *Machine Learning* tem ganhado destaque, especialmente na personalização da educação, ao adaptar o conteúdo e as metodologias de ensino ao perfil de cada aluno. Esse movimento é parte de uma tendência maior em direção à Educação 4.0, que visa integrar a tecnologia de maneira mais profunda nas salas de aula, promovendo um aprendizado mais interativo e centrado no aluno.

O ensino da matemática no nível básico, em particular, apresenta desafios únicos que podem ser mitigados com o uso de tecnologias inteligentes, visto que o baixo desempenho dos alunos nessa disciplina tem sido uma preocupação constante entre educadores e formuladores de políticas públicas, motivando a busca por novas abordagens que tornem o aprendizado mais acessível e eficaz. Nesse sentido, os Sistemas Tutores Inteligentes (STI) surgem como uma solução promissora, utilizando técnicas de *Machine Learning* para criar ambientes de aprendizagem adaptativos, que respondem de maneira dinâmica às necessidades individuais dos estudantes.

Os Sistemas Tutores Inteligentes (STI), quando aplicados ao ensino da matemática no nível básico por meio de técnicas de *Machine Learning*, viabilizam a criação de ambientes de aprendizagem altamente dinâmicos e adaptativos, visto que esses ambientes são particularmente eficazes para promover o

desenvolvimento da autonomia dos estudantes, ao priorizar um modelo educacional centrado no aluno, no qual o estudante assume um papel mais ativo e participativo no processo de aprendizagem da matemática. Além disso, os STI podem aumentar significativamente o nível de interação dos alunos ao monitorar suas ações e oferecer *feedback* personalizado e adaptado em tempo real, ajustando-se continuamente às necessidades individuais de aprendizagem. Portanto, esses sistemas visam complementar e enriquecer a atuação docente na qualidade do ensino e da aprendizagem da matemática, proporcionando um suporte adicional que se alinha com os objetivos pedagógicos do curso e promove um ambiente de aprendizado mais eficaz e engajador, buscando contribuir para o avanço das práticas educacionais, oferecendo insights sobre como as tecnologias emergentes podem ser integradas ao currículo escolar de forma eficaz e acessível.

II. TRABALHOS RELACIONADOS

Podemos observar várias semelhanças e diferenças na abordagem pedagógica, metodológica e tecnológica quando comparamos com outros trabalhos de grande relevância.

A. Semelhanças

Objetivo Comum: Todos os trabalhos focam em sistemas de tutoria inteligentes voltados para melhorar o processo de ensino-aprendizagem por meio da personalização e apoio adaptativo ao aluno. O Tutor Inteligente utiliza *machine learning* para adaptar o conteúdo às necessidades individuais dos alunos, assim como o Cognitive Tutor e o sistema PAT (Practical Algebra Tutor), que monitoram o progresso do aluno e oferecem *feedback* personalizado.

Modelagem Cognitiva: No caso dos tutores cognitivos, há uma ênfase na modelagem dos processos cognitivos do estudante, como destacado em *Cognitive Tutors: Lessons Learned* [1]. Esse tipo de modelagem também é fundamental no Tutor Inteligente, onde algoritmos de aprendizado de máquina tentam capturar padrões de dificuldades dos alunos e ajustar as estratégias pedagógicas.

Feedback em Tempo Real: Tanto o Tutor Inteligente quanto os tutores cognitivos mencionados nos artigos [1] e [2] empregam *feedback* imediato para corrigir o caminho do aluno e fornecer ajuda relevante, o que tem um efeito positivo tanto na aprendizagem quanto na motivação dos alunos.

B. Diferenças

Abordagem Tecnológica: O Tutor Inteligente descrito utiliza algoritmos de *machine learning* modernos, como redes neurais e técnicas de regressão, enquanto os tutores desenvolvidos por [2] são baseados em princípios da teoria ACT (*Adaptive Control of Thought*) e modelagem de produção,

focados em regras cognitivas *if-then* [1]. Isso implica que o Tutor Inteligente pode ser mais flexível na adaptação a novos padrões de dados conforme mais informações são obtidas, enquanto os tutores cognitivos têm uma base teórica mais estruturada e rígida.

Contexto de Aplicação: O Tutor Inteligente é aplicado especificamente no ensino de matemática no ensino básico, enquanto o PAT foi projetado para o ensino médio, focando em problemas de álgebra prática. A abordagem do PAT envolve a resolução de problemas do mundo real com o uso de ferramentas computacionais, como planilhas e gráficos [1], enquanto o Tutor Inteligente pode se concentrar mais em problemas básicos de matemática, com um possível escopo mais limitado.

Evolução e Escalabilidade: Enquanto o Cognitive Tutor e o PAT foram validados em larga escala em várias escolas, envolvendo centenas de alunos e mostrando melhorias significativas em testes padronizados e situacionais [1], o Tutor Inteligente está em fase de desenvolvimento ou de experimentação, dependendo da amplitude da aplicação.

III. TUTOR INTELIGENTE

O conceito de Tutor Inteligente, aplicado ao ensino de matemática com Machine Learning, refere-se a um sistema que adapta o aprendizado conforme as necessidades de cada aluno. Por outro lado, quanto mais interações de comando e resposta os algoritmos tornam-se cada vez mais precisos e melhoram sua performance em proceder uma pesquisa e apresentar um resultado cada vez mais próximo da inteligência humana. [3]. A interatividade proporcionada por plataformas digitais e recursos multimídia aumenta o engajamento dos alunos, especialmente aqueles que possuem dificuldades de concentração em aulas tradicionais [4]. Além disso, ele identifica dificuldades de aprendizagem de forma precoce, permitindo intervenções pedagógicas precisas [5]. A implementação de tutores inteligentes oferece um modelo pedagógico centrado no aluno, ajustando ritmo e complexidade de acordo com o progresso individual. As ferramentas digitais permitem a visualização e manipulação de conceitos abstratos de forma que facilita a compreensão dos alunos [6]. Por fim, o Tutor Inteligente não substitui o professor, mas complementa o ensino, contribuindo para uma educação mais equitativa e acessível [7].

A. Processo de desenvolvimento

Adotou-se uma abordagem ágil utilizando o SCRUM para organizar as fases do projeto, priorizar tarefas e garantir entregas incrementais. A combinação com o Kanban permitiu um gerenciamento eficaz do fluxo de trabalho, visualizando o progresso das tarefas e facilitando ajustes contínuos conforme novas demandas surgiam.

A equipe, composta por mais de 20 alunos de cursos de Sistemas de Informação, Engenharia da Computação e Ciência da Computação de universidades do Rio de Janeiro, foi dividida em áreas como QA, Front-end, Back-end, DevOps, Jogos e Design. A equipe de QA utilizou o *framework* Cypress para testes *End-to-End* (E2E), enquanto o desenvolvimento do sistema foi feito com o *framework* Django.

B. Arquitetura e Tecnologias

A metodologia de desenvolvimento abordada TDD (*Test-Driven Development*) que é uma metodologia que consiste em que os testes são escritos antes do código, conforme as 3 etapas de composição do TDD exibidos conforme na Figura 1:

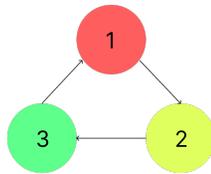


Fig. 1. Ciclo de desenvolvimento do TDD.

1. Escrever um teste: Antes de implementar uma nova funcionalidade, o desenvolvedor escreve um teste automatizado que define o que a funcionalidade deve fazer. Este teste inicialmente falha, já que o código correspondente ainda não foi implementado.

2. Implementar o código: Em seguida, o desenvolvedor escreve o código mínimo necessário para fazer o teste passar. A ideia é implementar apenas o suficiente para que o teste seja bem-sucedido.

3. Refatorar: Com o teste passando, o código é refatorado para melhorar a estrutura, legibilidade e eficiência, sem alterar o comportamento. Novos testes podem ser escritos conforme necessário, e o ciclo se repete.

O modelo de arquitetura utilizado no Tutor Inteligente foi o MVT (*Model-View-Template*), padrão do *framework* Django, que organiza a aplicação em três camadas: *Model* (lógica de negócios e estrutura de dados), *View* (intermediária que processa as requisições e interage com o *Model*) e *Template* (interface do usuário). Essa separação facilita o desenvolvimento e manutenção, garantindo que a lógica, apresentação e dados sejam geridos de forma independente.

Python foi escolhido pela sua flexibilidade, tanto no desenvolvimento web com Django quanto na criação de APIs via Django REST *framework*, além de ser usado nas Inteligências Artificiais integradas ao sistema distribuído APIIlo. JavaScript foi empregado no front-end para animações e chamadas de APIs. O banco de dados escolhido foi o PostgreSQL, conhecido

por sua robustez e confiabilidade. Bootstrap foi utilizado para construir interfaces responsivas.

Git e GitHub gerenciaram o versionamento de código e facilitaram a colaboração. GitHub Actions foi usado para pipelines de CI/CD, combinando testes End-to-End (E2E) e testes unitários escritos em TDD, minimizando bugs em produção. Cypress foi adotado para testes automatizados de interface gráfica.

Ollama, uma solução open-source para Modelos de Linguagem de Grande Escala (LLMs), foi usado no Gerador de Perguntas. Docker forneceu um ambiente padronizado, eliminando problemas de compatibilidade entre máquinas. Já o PyTorch está sendo utilizado para tarefas de visão computacional, como OCR, através do TorchVision para a segunda versão do *software*.

C. Informações sobre o desenvolvimento

O desenvolvimento do Tutor Inteligente teve início em Novembro de 2023, com uma equipe inicial de cinco alunos, que se dedicaram a uma carga horária de seis horas diárias, de segunda a sexta-feira. O trabalho foi realizado remotamente, utilizando um servidor no aplicativo *Discord*. No desenvolvimento do Tutor Inteligente, a criação e preparação do dataset incluiu filtragem e classificação de dados estudantis. Foram aplicadas técnicas de aprendizado de máquina, com destaque para a Clusterização com k-means, que obteve acurácia de 0.9878 (98 por cento) na segmentação de perfis. O uso de PyTorch para OCR e o módulo preditivo do FuzzyLab aprimoraram a personalização do sistema, adaptando o conteúdo às necessidades dos alunos. O código-fonte da aplicação foi compartilhado via GitHub, e o primeiro MVP (Produto Mínimo Viável) foi concluído em Dezembro do mesmo ano, com recursos voltados para a Língua Brasileira de Sinais (Libras), como a *Matriz de Confusão e Visualização de Clusterização com K-means* [Figura 2], *Calculadora de Libras* [Figura 3], *Tradutor de Libras* [Figura 4] e *Soletrador de Libras* [Figura 5].

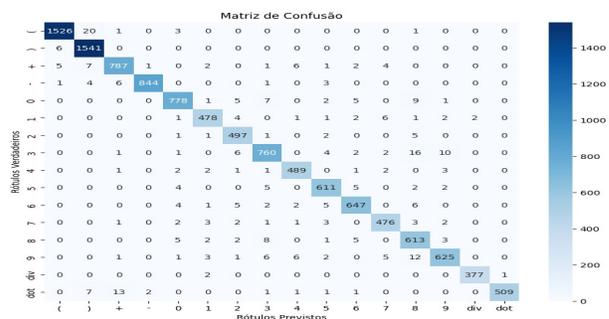


Fig. 2. Matriz de Confusão e Visualização de Clusterização com K-means

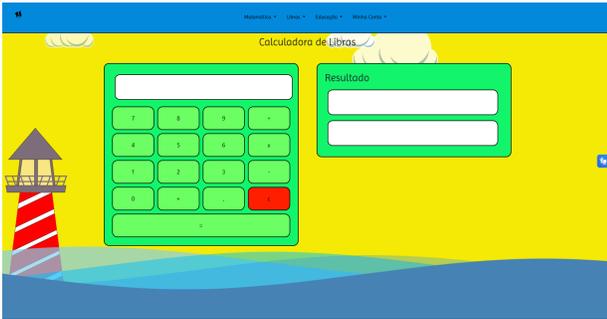


Fig. 3. Calculadora de Libras



Fig. 4. Tradutor de Libras



Fig. 5. Soletrador de Libras

Com o início do desenvolvimento do segundo MVP em janeiro de 2024, que se estendeu até agosto de 2024, a equipe cresceu para 15 pessoas, mantendo a mesma carga horária e a modalidade de trabalho remoto pelo *Discord*. Durante essa fase, foram desenvolvidos novos recursos significativos, incluindo:

IV. DISCUSSÃO E TRABALHOS FUTUROS

Os desafios enfrentados durante o desenvolvimento e a implementação do Tutor Inteligente destacaram várias áreas que necessitam de melhorias e de atenção futura, porém, uma das principais dificuldades é a adaptação dos alunos do ensino

básico ao uso de tecnologias de aprendizagem personalizadas, pois muitos alunos, especialmente aqueles com pouca familiaridade com o uso de plataformas digitais, têm apresentado dificuldades iniciais em interagir com o sistema, o que afeta o desempenho nas primeiras fases de uso.

Outro desafio significativo está relacionado à manutenção e à escalabilidade do sistema, pois conforme o número de usuários aumenta, surgem novas necessidades de aprimoramento da infraestrutura para garantir o funcionamento eficiente do tutor. Adicionalmente, a integração de funcionalidades adicionais, incluindo reconhecimento avançado de padrões mais complexos e a incorporação de técnicas mais avançadas de machine learning, requer um planejamento cuidadoso, tanto em termos de desenvolvimento de *software* quanto na formação de professores para lidar com essas novas ferramentas.

Para os trabalhos futuros, um dos principais focos será o desenvolvimento da versão 2.0 do Tutor Inteligente, que busca expandir as funcionalidades existentes, incluindo uma maior integração com outras disciplinas além da matemática, e a adição de novos módulos de aprendizado, como jogos interativos que auxiliam na fixação de conteúdos. Outro ponto importante será a avaliação contínua da eficácia do tutor, por meio de estudos longitudinais, que permitam acompanhar o impacto a longo prazo na performance dos alunos.

Além disso, será necessário explorar novas funcionalidades, como o suporte a alunos com necessidades especiais, através de interfaces mais acessíveis e recursos adaptados, como a integração de ferramentas de Libras, que já começou a ser implementada em fases iniciais. Esse avanço busca tornar o Tutor Inteligente uma solução mais inclusiva e adaptada à realidade de todos os alunos, independentemente de suas particularidades.

REFERÊNCIAS

- [1] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, "Cognitive tutors: Lessons learned," *The journal of the learning sciences*, vol. 4, no. 2, pp. 167–207, 1995.
- [2] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark, "Intelligent tutoring goes to school in the big city," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 30–43, 1997.
- [3] F. G. Cozman, G. A. Plonski, and H. Neri, *Inteligência artificial: avanços e tendências*. Instituto de Estudos Avançados, 2021, [livro eletrônico]. [Online]. Available: <https://www.livrosabertos.sibi.usp.br/portaldelivrosUSP/catalog/download/650/579/2181?inline=1>
- [4] C. A. Silva, "O impacto das tecnologias digitais no engajamento dos estudantes," *Revista de Tecnologia Educacional*, vol. 29, no. 4, pp. 56–67, 2017.
- [5] K. VanLehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educational psychologist*, vol. 46, no. 4, pp. 197–221, 2011.
- [6] J. A. Valente, "A utilização de ferramentas digitais no ensino da matemática," *Revista de Educação e Matemática*, vol. 45, no. 3, pp. 123–135, 2020.
- [7] B. P. Woolf, *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann, 2010.