



Ferramentas Visuais para o Ensino de *Machine Learning* na Educação Básica

Christiane Gresse von Wangenheim, UFSC, c.wangenheim@ufsc.br

Jean C. R. Hauck, UFSC, jean.hauck@ufsc.br

Fernando S. Pacheco, IFSC, fspacheco@ifsc.edu.br

Matheus F. Bertonceli Bueno, UFSC, matheus.felipe.bueno@grad.ufsc.br

Resumo: Ensinar *Machine Learning* (ML) em escolas ajuda a preparar estudantes para lidar com questões sociais, econômicas e éticas do século 21. Isso requer ferramentas apropriadas a cada idade que permitem que os estudantes criem uma concepção abrangente de ML e sua aplicação. Desse modo, apresentamos um mapeamento de ferramentas visuais emergentes que ajudam a ensinar conceitos de ML na Educação Básica e caracterizamos as ferramentas em relação ao suporte para o desenvolvimento e *deployment* de modelos de ML em contextos educacionais. Os resultados podem ajudar designers instrucionais e instrutores a selecionar ferramentas apropriadas a fim de contribuir para a democratização de competências em ML.

Palavras-Chave: *machine learning*; ferramenta visual; ensino de computação; educação Básica

Visual Tools for Teaching Machine Learning in K-12

Abstract: Teaching Machine Learning (ML) in school helps students to be better prepared to deal with the social, economic, and ethical issues in the 21st century. This requires age-appropriate tools that allow students to develop a comprehensive understanding of ML and its application. Therefore, we present a mapping of emerging visual tools that support the teaching of ML concepts in K-12 and we characterize the tools with respect to their support for the development of ML models as well as their deployment and their educational context. The results presented can help instructional designers and instructors to select appropriate tools for ML education in order to contribute to the democratization of ML competencies.

Keywords: Machine Learning; Visual tool; Computing Education; K-12

1. Introdução

Aprendizado de Máquina ou *Machine Learning* (ML) é um ramo da Inteligência Artificial focado no estudo de algoritmos que aprendem com os dados e melhoram com o tempo, sem serem programados de forma explícita para a solução de um problema específico. Atualmente ML está inserido em muitos dispositivos e serviços que fazem parte de nossa vida cotidiana e, portanto, entender conceitos básicos de ML torna-se importante para pessoas de todas as idades, incluindo crianças (Touretzky et al., 2019)(Wong et al., 2020). Apesar de ser uma área de conhecimento complexa, estudos têm mostrado que crianças são capazes de aprender conceitos de ML a partir de uma idade relativamente jovem e que a exposição a esses conceitos tem o potencial de elevar as suas habilidades cotidianas (Hitron et al., 2019). Isso apresenta novos desafios para o ensino de computação, como o de apresentar desde cedo conceitos de ML de modo que os alunos não se tornem somente consumidores, mas criadores de soluções inteligentes (Kahn e Winters, 2018).

O ensino de conceitos e técnicas fundamentais de ML tradicionalmente só tem sido feito no ensino superior (Torrey, 2012). Já existem diversos programas que focam no ensino de programação e robótica na Educação Básica, entretanto, ainda precisa ser abrangido o ensino da aplicação de ML (Hubwieser et al., 2015)(Touretzky et al., 2019). Para ser efetivo, um programa de ensino deve incluir desde a compreensão de conceitos básicos de ML até competências relacionadas à aplicação, ensinando os estudantes a criar modelos de ML na forma de “aprender fazendo”. Desse modo, ao vivenciarem as



possibilidades, pontos fortes e fracos dessa tecnologia, acredita-se que os estudantes possam alcançar uma compreensão mais profunda do tema (Wong et al., 2020)(Marques et al., 2020).

O desenvolvimento de aplicativos de ML não é trivial e o seu processo de desenvolvimento difere de um software tradicional, pois envolve adquirir um conjunto rotulado de exemplos, selecionar um algoritmo de aprendizado apropriado e seus parâmetros, treinar um modelo, avaliar as previsões desse modelo em relação ao conjunto de testes e, finalmente, sua implantação em produção (Ramos et al., 2020). Normalmente, os modelos de ML são desenvolvidos usando linguagens de programação baseadas em texto que requerem codificação. Para democratizar o aprendizado de ML, é desejável reduzir significativamente o esforço cognitivo, permitindo ao estudante se concentrar na lógica para resolver o problema. Observa-se uma curva acentuada de aprendizado envolvida no entendimento dos fundamentos de ML e das nuances sintáticas de várias bibliotecas utilizadas. Essas dificuldades são ainda mais salientes em algoritmos e bibliotecas para *Deep Learning* (DL). Assim, necessita-se de ferramentas intuitivas para criar, de forma mais fácil, modelos de ML em geral e também para DL (Tamilselvam et al., 2019).

Para esse fim, ao longo do tempo foram desenvolvidas ferramentas visuais que permitem aos usuários criar programas arrastando e soltando elementos gráficos, incluindo linguagens baseadas em blocos ou em fluxo, entre outras (Weintrop, 2019) (Burnett e Baker 1994). As vantagens dessas ferramentas visuais levaram a uma ampla adoção em contextos de programação introdutórios em diferentes estágios educacionais (Bau et al. 2017). Especialmente na Educação Básica, as linguagens de programação baseadas em blocos, como Scratch, SNAP!, Blockly e App Inventor, são proeminentes no ensino de algoritmos e programação (Weintrop, 2019). Seguindo esse sucesso, várias ferramentas visuais estão sendo propostas também para o ensino de ML, as quais são tipicamente formadas por um componente de desenvolvimento de modelos e um componente de *deployment* (Garcia et al., 2020). O componente de desenvolvimento de modelos de ML suporta a preparação de dados, permitindo coleta e rotulagem dos mesmos, a construção de modelos usando algoritmos de ML disponíveis, a avaliação de desempenho do modelo com dados de teste e, por fim, a exportação do modelo para uma plataforma de programação. Já o componente de *deployment* viabiliza o desenvolvimento de um sistema de software usando o modelo de ML criado previamente, permitindo a criação de soluções inteligentes que tornem o ensino da computação envolvente e motivador.

No entanto, não existe uma visão sistemática sobre quais ferramentas visuais existem para o ensino de ML no ensino fundamental e médio e quais são suas características. García et al. (2020) apresentam apenas uma comparação de algumas ferramentas, enquanto Hauck et al. (2019) enfocam ferramentas para desenvolver aplicações de *Internet of Things* com inteligência artificial. Outras revisões focam em aspectos diferentes, como pensamento computacional (Bordini et al., 2017)(Hubwieser et al., 2015). Portanto, apresentamos neste artigo os resultados de um estudo de mapeamento sistemático que podem ajudar designers instrucionais e educadores a escolher a ferramenta visual mais adequada, além de orientar pesquisadores na evolução dessas ferramentas.

2. Definição e Execução do Mapeamento

Para levantar o estado da arte, foi conduzido um mapeamento seguindo os procedimentos propostos por Petersen et al. (2008). Diferente de uma revisão sistemática da literatura, mapeamentos fornecem uma visão mais ampla de um tema enfocando na organização e



categorização dos objetos de estudo agregando as pesquisas existentes, ao invés de uma revisão profunda dos estudos individuais e da sua qualidade (Kitchenham et al., 2010)(Petersen et al., 2008).

2.1. Definição do protocolo de revisão

A pergunta de pesquisa é: Quais ferramentas visuais existem para ensinar conceitos de *Machine Learning* no contexto do ensino fundamental e ensino médio? Essa pergunta de pesquisa é refinada nas seguintes questões de análise:

AQ1. Quais ferramentas existem?

AQ2. Quais são as suas características em relação à plataforma de *ML*?

AQ3. Quais são as características delas em relação à plataforma de *deployment*?

AQ4. Como as ferramentas de *ML* foram desenvolvidas e avaliadas?

Critérios de inclusão/exclusão. Foram consideradas ferramentas para o desenvolvimento e *deployment* de modelos *ML* no contexto do ensino fundamental e médio publicadas entre 2010 e 2020. Incluímos também ferramentas que não foram exclusivamente desenvolvidas para este fim, porém cuja adoção neste nível educacional foi relatado. Ferramentas não abordando conceitos de *ML* e/ou não adotando um paradigma visual não foram consideradas. Excluímos também ferramentas somente voltadas para ensino infantil ou para uso profissional. Foram consideradas apenas ferramentas para as quais foram encontradas informações suficientes. Portanto, resumos ou artigos de apenas uma página foram excluídos.

Fontes dos dados. Foram examinados materiais e artigos publicados em inglês disponíveis nas principais bibliotecas digitais acessíveis por meio do Portal Capes. Foi realizada também uma busca no Google, a fim reduzir o risco de omissão de ferramentas não publicadas em bibliotecas científicas. A fim de minimizar ainda mais o risco de omissão, também incluímos literatura via *backward* e *forward snowballing* e literatura secundária como trabalhos acadêmicos referentes às ferramentas encontradas para obter maiores informações (Wohlin, 2014).

Definição da *Search String*. A *search string* foi composta de conceitos relacionados à questão de pesquisa, incluindo sinônimos. Dessas palavras-chave, a *search string* foi adaptada para cada fonte de dados apresentada na Tabela 1.

Tabela 1. *Search string* para cada fonte

Repositório	<i>Search string</i>
ACM Digital Library	(Abstract: "visual programming") OR (Abstract: "block-based programming") OR (Abstract: "gui tool") OR (Abstract: toolkit) AND ((All: "machine learning") OR (All: "neural network")) AND (Publication Date: (01/01/2010 TO 12/31/2020))
IEEE Xplore Digital Library	((("Abstract": "visual programming" OR "Abstract": "block-based programming" OR "Abstract": "gui tool" OR "Abstract": "toolkit") AND ("Abstract": "machine learning" OR "Abstract": "neural network")) Filters Applied: 2010 - 2020
Scopus	TITLE-ABS-KEY (("visual programming" OR "block-based programming" OR "gui tool" OR toolkit) AND ("machine learning" OR "neural network")) AND PUBYEAR > 2010 AND (LIMIT-TO (SUBJAREA , "COMP"))
Google	"block-based" "machine learning"

2.2. Execução da busca e extração dos dados

A pesquisa foi realizada em Junho-Julho de 2020 pelos autores (Tabela 2). Na primeira fase de análise, títulos e resumos foram analisados. No segundo estágio, os materiais



foram lidos por inteiro, para analisar sua relevância com respeito aos critérios de inclusão/exclusão.

Tabela 2. Número de artigos identificados por repositório e por fase de seleção

Fonte	No. de resultados da busca	No. de resultados analisados	No. de documentos potencialmente relevantes	No. de documentos relevantes
ACM	231	231	12	3
IEEE	269	269	8	2
SCOPUS	568	300	6	3
Google	460.000	300	11	4
<i>Backward snowballing</i>				6
<i>Forward snowballing</i>				3
Total de artigos relatando adoção de ferramentas de ML no ensino fundamental/médio				21 (sem duplicatas)

Adotando *backward e forward snowballing*, foram encontradas 9 ferramentas relevantes. Documentos duplicados foram eliminados e aqueles que descrevem a mesma ferramenta foram unificados.

3. Resultados

3.1. Quais ferramentas existem?

Como resultado, foram encontradas 15 ferramentas para suporte ao ensino de *ML* na Educação Básica.

Quadro 1. Ferramentas existentes

Nome	Descrição	Escopo		Referência
		Plataforma de ML	Plataforma de deploy	
AlpacaML	Um aplicativo iOS que oferece suporte na criação, teste, avaliação e uso de modelos de <i>ML</i> de gestos baseados em dados de sensores vestíveis.	x	Scratch	A. Zimmermann-Niefield et al. 2020. Youth making machine learning models for gesture-controlled interactive media. In Proc. of the Interaction Design and Children Conference. ACM, 63–74. A. Zimmermann-Niefield et al. 2019a. Sports and machine learning: How young people can use data from their own bodies to learn about machine learning. XRDS, 25(4), 44–49. A. Zimmermann-Niefield et al. 2019b. Youth Learning Machine Learning through Building Models of Athletic Moves. In Proc. of the 18th International Conference on Interaction Design and Children. ACM, 121–132.
BlockWiSARD	Um ambiente de programação visual que utiliza o WiSARD WANN para permitir que as pessoas desenvolvam sistemas com alguma capacidade de aprendizado.	x	BlockWiSARD	R. L. Queiroz et al. 2020. AI from concrete to abstract: demystifying artificial intelligence to the general public. arXiv:2006.04013 [cs.CY].
Cognimates	Uma plataforma educacional de IA para programar e personalizar o desenvolvimento de modelos de IA incorporados em dispositivos, como Alexa e Cozmo.	x	Scratch	S. Druga. 2018. Growing up with AI: Cognimates: from coding to teaching machines. Master thesis, MIT, USA. S. Druga et al. 2019. Inclusive AI literacy for kids around the world. In Proc. of FabLearn. ACM, 104–111. http://cognimates.me/home/
eCraft2learn	Blocos adicionais para Snap! que fornecem uma interface fácil de usar para os serviços de nuvem de IA e a funcionalidade de <i>deep learning</i> .	x	Snap!	K. M. Kahn e N. Winters. 2018. AI Programming by Children. In Proc. of the Conference on Constructionism, Vilnius, Lithuania. K. M. Kahn et al. 2018. AI Programming by Children Using Snap! Block Programming in a Developing Country. In Proc. of the 13th European Conference on Technology Enhanced Learning, Leeds, UK, 2018. K. M. Kahn et al. 2020. Deep learning programming by all. In Proc. of the Conference on Constructionism, Dublin, Ireland. K. M. Kahn e N. Winters. 2017. Child-Friendly Programming Interfaces to AI Cloud Services. In: Lavoué É. et al. (eds) Data Driven Approaches in



				Digital Education. Lecture Notes in Computer Science, vol 10474. Springer, Cham. https://ecraft2learn.github.io/ai/
Educational Approach to ML with Mobile Applications	Um conjunto de extensões do App Inventor que abrangem vários subcampos de ML, entre os quais a extensão <i>Teachable Machine</i> permite desenvolver um modelo de ML.	x	App Inventor	K. Zhu. 2019. An Educational Approach to Machine Learning with Mobile Applications. M.Eng thesis, MIT, Cambridge, MA, USA.
Google Teachable Machine	Uma interface baseada na Web que permite que as pessoas treinem seus próprios modelos de classificação ML sem codificar, usando sua webcam, imagens ou som.	x	-	M. Carney et al. 2020. Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification. In Proceedings of Conference on Human Factors in Computing Systems, ACM https://teachablemachine.withgoogle.com/
LearningML	Uma plataforma destinada a aprender ML supervisionado no ensino fundamental e médio.	x	Scratch	J. D. Rodríguez-García et al. 2020. LearningML: A Tool to Foster Computational Thinking Skills Through Practical Artificial Intelligence Projects. Distance Education Journal, 20(63). https://learningml.org/
mBlock	Um software de programação baseado em código textual e em blocos e que permite criar um modelo de ML a partir de suas extensões de IA e para <i>Teachable Machine</i> .	x	mblock	https://www.mblock.cc https://www.mblock.cc/en-us/blog/mblock/update-ai-axis
Milo	Um ambiente de programação visual baseado na Web para o ensino de ciência de dados.	x	Milo	A. Rao et al. 2018. Milo: A visual programming environment for Data Science Education. In Proc. of the Symposium on Visual Languages and Human-Centric Computing, Lisbon, Portugal, 211-215.
ML4K	Uma ferramenta que introduz ML fornecendo experiências práticas para o treinamento de sistemas de ML e a criação de artefatos.	x	Scratch, App Inventor, Python	D. Lane. 2018. Explaining Artificial Intelligence. Hello World, 4. https://machinelearningforkids.co.uk/
Orange	Um kit de ferramentas de visualização e mineração de dados e ML para análise exploratória de dados e visualização interativa de dados.	x	-	J. Demšar et al. 2013. Orange: data mining toolbox in Python. Journal of Machine Learning Research, 14, 2349-2353. P. Godec et al. 2019. Democratized image analytics by visual programming through integration of deep models and small-scale machine learning. Nature Communications, 10, Article 4551. https://orange.biolab.si/
Personal Image Classifier	Um sistema da web para treinar, testar e analisar modelos de classificação de imagens com uma extensão para o MIT App Inventor que permite usar os modelos em aplicativos móveis.	x	App Inventor	D. Tang et al. 2019. PIC: A Personal Image Classification Webtool for High School Students. In Proc. of the IJCAI EduAI Workshop, Macao, China. D. Tang. 2019. Empowering Novices to Understand and Use Machine Learning With Personalized Image Classification Models, Intuitive Analysis Tools, and MIT App Inventor, M.Eng thesis, MIT, Cambridge, USA. https://classifier.appinventor.mit.edu/
RapidMiner	Plataforma abrangente de ciência de dados com design visual de fluxo de trabalho e automação completa de soluções ML.	x	RapidMiner Studio, RapidMiner AI Hub, JupyterLab	B. Sakulkeakulsuk et al. 2018. Kids making AI: Integrating Machine Learning, Gamification, and Social Context in STEM Education. In Proc. of the Int. Conference on Teaching, Assessment, and Learning for Engineering, Wollongong, Australia, 1005-1010. https://rapidminer.com
Scratch Nodes ML	Um sistema que permite criar e compartilhar reconhecedores de gestos personalizados.	x	Scratch	A. Agassi et al. 2019. Scratch Nodes ML: A Playful System for Children to Create Gesture Recognition Classifiers. In Proc. of the Conference on Human Factors in Computing Systems, ACM, 1-6.
SnAlp	Uma estrutura que permite o aprendizado construcionista do <i>reinforcement learning</i> com o SNAP!	x	SNAP!	S. Jatzlau et al. 2019. It's not Magic After All – Machine Learning in Snap! using Reinforcement Learning. In Proc. of IEEE Blocks and Beyond Workshop, Memphis, TN, USA, 37-41. https://ddi.cs.fau.de/schule/snaip/

A maioria das ferramentas suporta de forma abrangente tanto o desenvolvimento de modelos de ML quanto o seu *deployment* em ambientes de programação tipicamente utilizados na Educação Básica, como Scratch, App Inventor e Snap!. A maioria das



ferramentas está disponível online gratuitamente, porém algumas necessitam de cadastro do usuário e/ou uso de *API keys*, o que pode confundir o público alvo. A maioria das ferramentas está disponível somente em inglês. Somente mBlock, ML4K, BlockWiSARD estão disponíveis também em português.

Seguindo a estratégia de aprender fazendo, as ferramentas são tipicamente utilizadas em unidades extracurriculares de curta duração (1-6 horas). Embora existam unidades voltadas para crianças menores (a partir dos 6 anos), muitas se concentram no ensino médio ((Rao e t al., 2018)(Tang et al. 2019)(Jatzlau et al., 2019)(Zhu, 2019)). As ferramentas são acompanhadas por diversos graus de suporte instrucional, incluindo, principalmente, tutoriais passo a passo para as atividades práticas usando as ferramentas.

Considerando o desconhecimento sobre ML por parte desse público alvo, todas as ferramentas são direcionadas principalmente a novatos nessa área. No entanto, fornecendo um modo avançado, algumas das ferramentas também permitem que usuários mais experientes interajam em um nível mais detalhado ao construir, treinar e/ou avaliar modelos de ML. Diversas ferramentas pressupõem que os usuários já tenham experiências anteriores com o respectivo ambiente de programação para *deployment*, por exemplo, por meio de cursos de programação.

3.2. Quais são as suas características em relação à plataforma de ML?

Praticamente todas as ferramentas se concentram na tarefa de reconhecimento de imagens ou fala (Figura 1), adotando o aprendizado supervisionado. Somente o SnAIp permite o *reinforcement learning* como parte de jogos com Snap!.

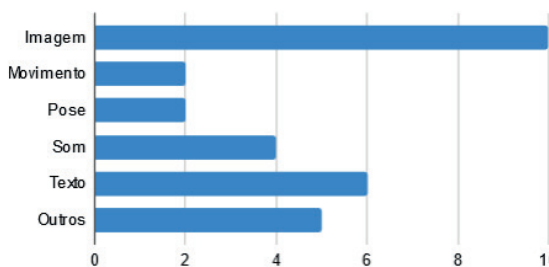


Figura 1. Frequência de tipo de dados suportados pelas ferramentas

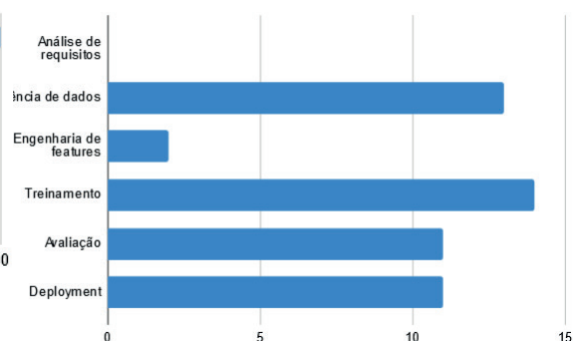


Figura 2. Frequência de etapas do processo de ML suportados

As ferramentas suportam grande parte do processo de desenvolvimento de *ML* (Figura 2). Primeiramente, suportam a coleta de pequenas quantidades de dados (via *upload*, webcam, microfone, etc.) e sua rotulagem, organizando-as em categorias. As ferramentas permitem o uso de uma variedade de tipos de dados, principalmente imagens. No entanto, poucas fornecem um acesso fácil a conjuntos de dados pré-existentes para auxiliar nesta etapa em sala de aula.

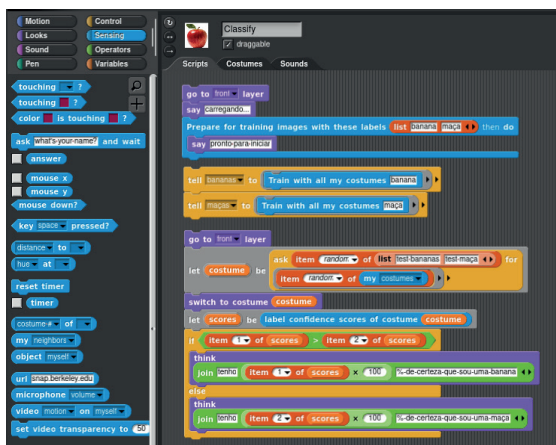
Algumas ferramentas permitem o uso desses dados para treinamento de um modelo por meio de *transfer learning*, o que permite construir modelos precisos de maneira mais fácil, usando diversos *back-ends* de *ML*, como IBM Watson, Keras ou Tensorflow. Em geral, as ferramentas suportam aprendizagem supervisionada, com poucas exceções suportando aprendizagem de *reinforcement* e/ou aprendizagem não supervisionada. O treinamento do modelo pode ser realizado na máquina local, algumas ferramentas permitem o uso de um servidor em nuvem ou diretamente em um dispositivo



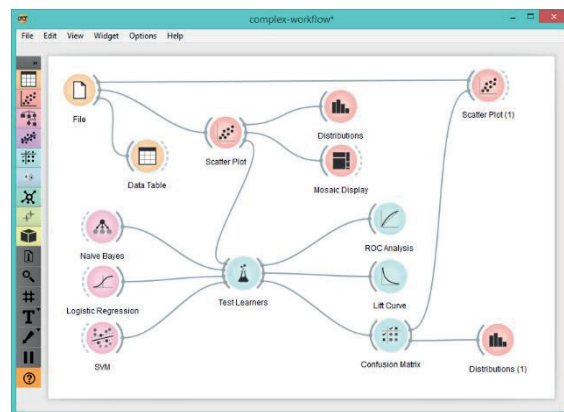
móvel. A maioria usa o navegador do usuário para treinar o modelo (Teachable Machine, PIC, LearningML, mBlock). Como o processo de treinamento do modelo às vezes pode ser demorado, as ferramentas que permitem o treinamento localmente podem fazer uso da GPU da máquina local, quando disponível, para acelerar o processo de treinamento (eCraft2Learn e RapidMiner).

Em uma terceira etapa, o desempenho do modelo é avaliado. De modo geral, o usuário pode testar o modelo com novos dados e, como saída, tem-se o rótulo resultante da predição ao qual a entrada pertence e o valor de confiança que representa a probabilidade de ocorrência. Algumas ferramentas também permitem a visualização de medidas de desempenho, como precisão e função de perda (Carney et al., 2020) ou a tabela de acurácia e matriz de confusão (Tang et al., 2019). A maioria das ferramentas suporta a exportação do modelo criado para *deployment* diretamente em um ambiente de programação baseado em blocos ou em diversos formatos (como Tensorflow.js ou Python).

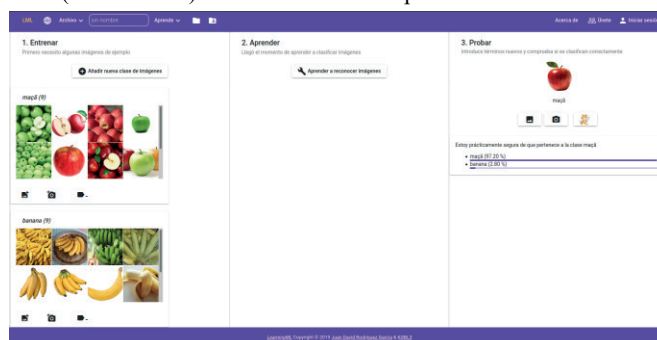
Foram identificados três tipos de suporte de ferramentas visuais para o desenvolvimento de modelos de *ML* (Figura 3). Alguns, como BlockWiSARD, Milo, ScratchNodes ML e SnAIp, fornecem suporte baseado em blocos de *ML*, estendendo o respectivo ambiente de programação. Outros, como Orange ou o RapidMiner adotam uma abordagem baseada em fluxo de dados, usando caixas, conectadas por setas, linhas ou arcos que representam relações direcionadas. Outros ainda adotam uma abordagem baseada em fluxo de trabalho que suporta o desenvolvimento, guiando o usuário passo a passo por meio de um navegador Web ((Carney et al., 2020)(García et al., 2020)(Tang et al., 2019)(Druga et al., 2019)) ou via interface visual baseada em aplicativo (Zimmermann-Niefeld et al., 2019).



Exemplo baseado em blocos (eCraft2learn)



Exemplo baseado em fluxo de dados (Orange)



Exemplo baseado em fluxo de trabalho (LearningML)

Figura 3. Exemplos de suporte de desenvolvimento de modelos *ML*

Seguindo a estratégia de reduzir a carga cognitiva, enquanto oferecendo espaço para o uso de conceitos mais complexos (piso baixo e teto alto) e permitindo diversos caminhos de exploração (paredes largas) (Resnick et al., 2000), essas ferramentas tendem a abstrair operações importantes durante o processo de aprendizagem, por exemplo, reduzindo o processo de treinamento do modelo à ação de um único botão. No entanto, essa ocultação de conceitos de *ML* limita a capacidade das pessoas de construir modelos mentais precisos, o que parece resultar não apenas em um menor efeito de aprendizado, mas aprendizado nulo (Resnick et al., 2000)(Hitron et al., 2019). Com esse objetivo, algumas ferramentas ((Demšar et al., 2013)(Tang et al. 2019)(Jatzlau et al., 2019)(Carney et al., 2020)) oferecem modos avançados que permitem a configuração de hiperparâmetros para treinamento (como taxa de aprendizado, épocas, etc.) ou medidas de avaliação mais detalhadas.

3.3. Quais são suas características com respeito à plataforma de *deployment*?

Enquanto algumas ferramentas permitem apenas a exportação do modelo de *ML* criado, várias também fornecem suporte para a sua aplicação como parte de um jogo, aplicativo móvel, etc., integrado ou como extensão de um ambiente de programação baseado em blocos (Figura 4).



Figura 4. Exemplo de suporte para o *deployment*

Dependendo das tarefas suportadas pela ferramenta, são fornecidos blocos de programação de *ML* com predominância para Scratch e App Inventor (Tabela 3). Dessa forma, as ferramentas permitem que os estudantes aprendam conceitos de *ML*, enquanto os capacitam a criar artefatos significativos com impacto direto em suas vidas e comunidades. Isso pode motivá-los a criar aplicações inovadoras que correspondam aos seus interesses e paixões, fornecendo uma ferramenta de aprendizado com paredes amplas (Kahn e Winters, 2018).

3.4. Como as ferramentas de *ML* foram desenvolvidas e avaliadas?

De maneira geral, as publicações encontradas carecem da descrição da metodologia de pesquisa adotada para o desenvolvimento das ferramentas de *ML*. Apenas Queiroz et al. (2020) indica o alinhamento com o construtivismo, construcionismo e construção do conhecimento e agentes inteligentes, processo de aprendizado e percepção da inteligência. Druga et al. (2019) indicam o uso de uma abordagem de design participativo com *codelab sprites*. O código fonte de várias ferramentas ((Druga et al., 2019)(Kahn e Winters, 2018)(Carney et al., 2020)(García et al., 2020)(Rao et al., 2018)(Lane, 2018)(Demšar et al., 2013)) (Tang et al., 2019)) está disponível sob licenças de código aberto, permitindo sua evolução.

No entanto, há vários relatos de estudos visando a avaliação das ferramentas. Os fatores avaliados variam desde a eficácia, usabilidade, utilidade e eficiência das



ferramentas na aprendizagem dos alunos além dos seus pontos fortes e fracos. As avaliações foram conduzidas como estudos de caso ou relatam aplicações de maneira informal (Carney et al., 2020). O tamanho das amostras é geralmente pequeno, variando de 6 a 23 participantes; apenas Kahn e Winters (2018) apresentam um estudo com 40 alunos, Sakulkueakulsuk et al. (2018) de 84 alunos e Druga et al. (2019) um grande estudo internacional replicado, incluindo 102 crianças. As descobertas desses estudos indicam que as ferramentas ajudam a alavancar o conhecimento do domínio dos alunos para coletar dados, construir, testar e avaliar modelos, permitindo que eles realizem iterações rápidas para testar hipóteses sobre o desempenho do modelo e a sua reformulação. A usabilidade das ferramentas também foi avaliada como muito boa. Além disso, as ferramentas ajudam os alunos a desenvolver e discutir teorias sobre como o modelo funciona e as características de um bom modelo, ajudando-os a compreender conceitos complexos de *ML*, bem como a refletir criticamente sobre seu impacto na prática. A integração em ambientes gerais de programação baseados em blocos também permitiu a criação de artefatos atraentes e úteis.

3.5. Ameaças à validade

A fim de minimizar as ameaças à validade dos resultados deste estudo, identificamos ameaças potenciais e aplicamos estratégias de mitigação. Revisões sistemáticas sofrem com o viés comum de que resultados positivos têm mais probabilidade de serem publicados do que negativos. No entanto, não consideramos isso uma ameaça crítica para nossa pesquisa, pois, em vez de focar no impacto dessas ferramentas, nosso objetivo era caracterizar as próprias ferramentas. Para mitigar a omissão de estudos relevantes, construímos cuidadosamente a *search string* para ser o mais inclusiva possível, considerando não apenas os conceitos centrais, mas também sinônimos. O risco de excluir estudos primários relevantes foi ainda mitigado pelo uso de vários bancos de dados e pela inclusão de literatura secundária. Foi realizada também a busca via Google de forma anônima para encontrar outras fontes de literatura. Considerando que a busca não foi somente realizada nesta base, esta inclusão é considerada complementar, a fim de minimizar o risco de omissão de ferramentas não publicadas por meio de artigos científicos (Hagstrom et al., 2015)(Haddaway et al., 2015). As ameaças à seleção e à extração de dados foram atenuadas, fornecendo uma definição detalhada dos critérios de inclusão/exclusão. Definimos e documentamos um protocolo rígido para a seleção dos estudos e todos os autores realizaram-na juntos, discutindo-a até que um consenso fosse alcançado.

4. Conclusão

Como resultado do mapeamento, identificamos um total de 15 ferramentas voltadas principalmente a iniciantes no nível do ensino médio. A maioria das ferramentas suporta tarefas de reconhecimento, fornecendo suporte para o processo completo de *ML*, desde a preparação dos dados até a avaliação usando diferentes tipos de linguagens visuais. Várias ferramentas são integradas em linguagens de programação baseadas em blocos, permitindo a integração direta dos modelos de *ML* criados como parte de soluções de software inteligentes. Os resultados de estudos empíricos indicam a usabilidade e utilidade dessas ferramentas na Educação Básica. No entanto, também observou-se uma falta de metodologia científica mais rigorosa para o desenvolvimento das ferramentas e sua funcionalidade, além de estudos mais detalhados para analisar de forma mais abrangente e sistemática o impacto dessas ferramentas na aprendizagem de *ML* na Educação Básica.



Agradecimentos

Este trabalho foi apoiado pelo CNPq, entidade do governo brasileiro focada no desenvolvimento científico e tecnológico.

Referências

- BAU, D., GRAY, J., KELLEHER, C., SHELDON, J., TURBAK, F. Learnable programming: blocks and beyond. **Communications of the ACM**, 60(6), 2017, 72-80.
- BORDINI, A., AVILA, C., MARQUES, M., FOSS, L., CAVALHEIRO, S. Pensamento Computacional nos Ensinos Fundamental e Médio: uma revisão sistemática. In: **Simpósio Brasileiro de Informática na Educação**, Recife, Brasil, 2017.
- BURNETT, M. M. & BAKER, M. J. A. Classification System for Visual Programming Languages. **Journal of Visual Languages and Computing**, 5, 1994, 287-300.
- HADDAWAY, N. R., COLLINS, A. M., COUGHLIN, D., KIRK, S. The role of Google Scholar in evidence reviews and its applicability to grey literature searching. **PLoS ONE**, 10(9), 2015, e0138237.
- HAGSTROM, C., KENDALL, S., CUNNINGHAM, H. Googling for grey: Using Google and Duckduckgo to find grey literature. In: **23rd Cochrane Colloquium**, 2015, 1–327.
- HAUCK, M., MACHHAMER, R., CZENKUSCH, L., GOLLMER, K. U., DARTMANN, G. Node and Block-Based Development Tools for Distributed Systems With AI Applications. **IEEE Access**, 7, 2019, 143109-143119.
- HITRON, T., ORLEV, Y., WALD, I., SHAMIR, A., EREL, H., ZUCKERMAN, O. Can Children Understand Machine Learning Concepts? The Effect of Uncovering Black Boxes. In: **Conference on Human Factors in Computing Systems**. ACM, New York, NY, USA, Paper 415, 2019, 1–11.
- HUBWIESER, P. ET AL. A Global Snapshot of Computer Science Education in K-12 Schools. In: **ITiCSE on Working Group Reports**, Vilnius, Lithuania, 2015.
- KITCHENHAM, B., BRERETON, P., BUDGEN, D. The educational value of mapping studies of software engineering literature. In: **32nd Int. Conference on Software Engineering**. IEEE, 2010, 589-598.
- MARQUES, L. S., GRESSE VON WANGENHEIM, C., HAUCK, J. C. R. Teaching Machine Learning in School: A Systematic Mapping of the State of the Art. **Informatics in Education**, 19(2), 2020.
- PETERSEN, K. FELDT, R., MUJTABA, S., MATTSSON, M. Systematic mapping studies in software engineering. In: **12th Int. Conference on Evaluation and Assessment in Software Engineering**, Bari, Italy, 2008, 68–77.
- RAMOS, G. MEEK, C., SIMARD, P., SUH, J., GHORASHI, S. Interactive machine teaching: a human-centered approach to building machine learned models”. **Human-Computer Interaction**, 2020.
- RESNICK, M., BERG, R., EISENBERG, M. Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. **The Journal of the Learning Sciences**, 9(1), 2000, 7--30.
- TAMILSELVAM, S. G. ET AL. A visual programming paradigm for abstract deep learning model development. In: **10th Indian Conference on Human-Computer Interaction**. ACM, New York, NY, USA, Article 16, 2019, 1–11.
- TORREY, L. Teaching Problem-Solving in Algorithms and AI. In: **3rd Symposium on Educational Advances in Artificial Intelligence**, Toronto, Canada, 2012.
- TOURETZKY, D. S. ET AL. K-12 Guidelines for Artificial Intelligence: What Students Should Know. In: **ISTE Conference**, Philadelphia, PA, USA, 2019.
- WEINTROP, D. Block-based Programming in Computer Science Education. **Communications of the ACM**, 62(8), 2019, 22-25.
- WOHLIN. C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: **18th Int. Conference on Evaluation and Assessment in Software Engineering**. ACM, Article 38, 2014.
- WONG, G. K. W., MA, X., DILLENBOURG, P., HUAN, J. Broadening artificial intelligence education in K-12: where to start? **ACM Inroads** 11(1), 20, 2020.