UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
SISTEMAS DE INFORMAÇÃO

Gustavo de Castro Salvador

Desenvolvimento de um Modelo de Avaliação Automatizada de Aprendizagem de *Machine Learning* voltado a Classificação de Imagens no Ensino Médio

Florianópolis
2021

Gustavo de Castro Salvador

Desenvolvimento de um Modelo de Avaliação de Aprendizagem de *Machine Learning* voltado a Classificação de Imagens no Ensino Médio

Trabalho de Conclusão do Curso de Graduação em Sistemas de Informação do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Sistemas de Informação
Orientadora: Prof.ª Dr.ª rer. nat. Christiane Gresse von Wangenheim, PMP

Florianópolis
2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Salvador, Gustavo de Castro
Desenvolvimento de um Modelo de Avaliação Automatizada de Aprendizagem de Machine Learning voltado à Classificação de Imagens no Ensino Médio / Gustavo de Castro Salvador ; orientadora, Christiane Gresse von Wangenheim, coorientador, Jean Carlo Rossa Hauck, 2021.
152 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Sistema de Informação, Florianópolis, 2021.

Inclui referências.

1. Sistema de Informação. 2. Machine Learning. 3. Avaliação de aprendizagem. 4. Classificação de imagens. 5. Ensino Médio. I. Gresse von Wangenheim, Christiane. II. Carlo Rossa Hauck, Jean. III. Universidade Federal de Santa Catarina. Graduação em Sistema de Informação. IV. Título.

Gustavo de Castro Salvador

Desenvolvimento de um Modelo de Avaliação de Aprendizagem de *Machine Learning* voltado a Classificação de Imagens no Ensino Médio

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel e aprovado em sua forma final pelo Curso de Sistemas de Informação.

Florianópolis, 27 de setembro de 2021.

_____
Prof. Cristian Koliver, Dr.
Coordenador do Curso

Banca Examinadora:

_____
Prof.ª Dr.ª rer. nat. Christiane Gresse von Wangenheim, PMP
Orientadora
Universidade Federal de Santa Catarina

_____
Prof. Dr. Jean Carlo Rossa Hauck
Coorientador
Universidade Federal de Santa Catarina

_____
Prof.ª MSc. Nathalia da Cruz Alves
Avaliadora
Instituto Federal de Santa Catarina

Este trabalho é dedicado a todos que veem a educação como a base para a construção de um mundo melhor.

RESUMO

Machine Learning (ML) está se tornando cada vez mais presente na realidade, com seus algoritmos tendo papel essencial em sistemas inteligentes que executam tarefas como reconhecimento de voz e recomendações personalizadas. A capacidade de criar soluções inteligentes com ML é uma característica cada vez mais requisitada para os profissionais do presente e futuro próximo, tornando-se essencial a inclusão de conteúdos de ML nos currículos de estudantes da Educação Básica. Embora já existam algumas unidades instrucionais para o ensino de ML, a avaliação da aprendizagem dos conceitos de ML continua sendo uma questão aberta e pouco abordada nos trabalhos existentes. Assim, o presente trabalho busca definir sistematicamente um modelo de avaliação de aprendizagem de estudantes, com foco no público do Ensino Médio para a tarefa de classificação de imagens. Com base na revisão sistemática de literatura, o modelo conceitual é criado sistematicamente seguindo o método de *Evidence-Centered Assessment Design* e automatizado no contexto de criação de modelos de ML usando Jupyter Notebook no ambiente do Google Colab com Python. O modelo de avaliação é aplicado em um curso de ML voltado à classificação de imagens de árvores nativas de Santa Catarina. A apresentação visual dos resultados é realizada no Jupyter Notebook, utilizando-se da biblioteca *ipywidgets*. Uma avaliação preliminar indicou a utilidade, funcionalidade e usabilidade do modelo. Deste modo, o presente trabalho traz uma solução inicial para automatizar a avaliação da aprendizagem do aluno com base no modelo de ML criado para Classificação de Imagens, visando realizar uma contribuição para o ensino de ML no Ensino Médio.

Palavras-chave: *Machine Learning*, Avaliação de aprendizagem, Classificação de imagens, Ensino Médio, Avaliação de desempenho

ABSTRACT

Machine Learning (ML) is becoming increasingly present in daily lives, playing an essential role in intelligent systems that perform tasks such as speech recognition and personalized recommendations. The ability to create intelligent solutions with ML is an increasingly required ability for professionals in the present and near future, making it essential to include ML content in K-12 students' curricula. Although there are some instructional units for teaching ML, ML concepts assessment remains an open question and with little attention given in existing works. The present work seeks to systematically define a learning assessment model for students, focusing on the High School audience for the image classification task. Based on a systematic literature review, the conceptual model is created systematically following the Evidence-Centered Design method and automated in the context of creating ML models using Jupyter Notebooks in the Google Colab environment with Python. The assessment model is applied in a ML course aiming at classifying images of native trees in Santa Catarina/Brazil. The visual presentation of results is performed in the Jupyter Notebook using the *ipywidgets* library. A preliminary evaluation indicated the model's usefulness, functionality and usability. Thus, the work presents an initial solution to automate the assessment of student learning based on the ML model created for Image Classification task, aiming to make a contribution to the teaching of ML in High School.

Keywords: Machine Learning, Learning assessment, Image classification, High school, Performance-based assessment

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BNCC Base Nacional Comum Curricular

DL *Deep Learning*

ECD *Evidence-Centered Assessment Design*

IA Inteligência Artificial

ML *Machine Learning*

SBC Sociedade Brasileira de Computação

SUS *System Usability Scale*

TI Tecnologias da Informação

TCT Teoria Clássica dos Testes

TRI Teoria de Resposta ao Item

TL *Transfer Learning*

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Aprendizado de máquina ou *Machine Learning* (ML) tornou-se parte da vida cotidiana, impactando profundamente nossa sociedade. Ele permite que os sistemas aprendam e melhorem automaticamente com suas experiências sem serem programados de maneira explícita, servindo como motor de inovação para uma ampla gama de aplicativos, desde sistemas de reconhecimento de voz a assistentes inteligentes, carros autônomos, etc. No entanto, a maioria das pessoas não entende a tecnologia por trás disso, o que pode tornar o ML um assunto misterioso ou até assustador, ofuscando seu potencial de impactar positivamente a sociedade (EVANGELISTA et al., 2018; HO, SCADDING, 2019).

Assim, para desmistificar o ML é importante introduzir conceitos e práticas básicas já na escola, permitindo que os estudantes se tornem não apenas consumidores, mas também criadores de soluções inteligentes (TOURETZKY et al., 2019a; KANDLHOFER et al., 2016). A exposição a este tipo de conhecimento complexo também tem o potencial de melhorar as habilidades cotidianas dos estudantes, fornecendo-lhes conhecimento para lidar com as questões sociais, econômicas e éticas criadas pelo uso de ML nas diversas esferas da sociedade (KAHN et al., 2020). Além disso, pode encorajar alunos a considerarem carreiras em computação e fornecer uma preparação sólida para o Ensino Superior, já que todos os estudantes possuirão uma vida altamente influenciada pela computação e muitos trabalharão em áreas em que a computação está diretamente envolvida (CSTA, 2011).

De acordo com AI4K12 (TOURETZKY et al., 2019a), como parte de umas das grandes ideias de aprendizagem de Inteligência Artificial, o ensino de ML nesta fase educacional deve incluir uma compreensão dos conceitos básicos de ML, como algoritmos de aprendizagem e fundamentos de redes neurais, assim como limitações e preocupações éticas relacionadas ao ML. Para tornar o ML mais claro e permitir que os alunos construam modelos mentais corretos, é importante incentivar a aprendizagem ativa, que enfatiza o fazer e a experiência direta dos alunos (WONG

et al., 2020). Como parte do desenvolvimento centrado no humano de um modelo de ML, os alunos podem explorar diversas tarefas da construção de modelos, desde a preparação de conjunto de exemplos, seleção de algoritmos de aprendizagem apropriados, treino de modelos e avaliação do desempenho de modelos (LWAKATARE et al., 2019; RAMOS et al., 2020, GRESSE VON WANGENHEIM, 2021).

Ainda assim, como ML se trata de uma área complexa, uma forma eficaz de abordagem seria começar formando competências de nível simples primeiro, depois progredir para uma maior complexidade. Portanto, o ciclo "*Use-Modify-Create*" (LYTLE et al., 2019), usado constantemente para a progressão na aprendizagem de computação, também pode ser adotado para o ensino de ML. Seguindo este ciclo, os alunos aprendem os tópicos de ML primeiro usando e analisando um determinado artefato de ML, depois modificando um existente, até eventualmente criar artefatos inteligentes.

Israel (GAL-ZER et al., 1995) e países do leste europeu (SYSLO, 2011; DAGIENE, 2008) já vêm oferecendo cursos de computação em suas escolas há décadas. Entretanto, apenas recentemente diversos países alteraram o foco do ensino de computação no Ensino Fundamental e Médio. Saindo de um foco de uso de aplicativos de computador e Tecnologias da Informação e Comunicação para ensinar computação, levando os alunos a se tornar criadores de Tecnologias da Informação (TI), demonstrando fluência em TI (HUBWIESER et al., 2015).

O *Machine Learning*, como um subcampo da computação, tem sido tradicionalmente ensinado apenas no Ensino Superior (TORREY, 2012; MCGOVERN et al., 2011). Embora hoje existam muitas iniciativas que têm como foco programação e robótica, o Ensino Fundamental e Médio ainda precisam abraçar o ensino de Inteligência Artificial, incluindo ML (HUBWIESER et al., 2015). No entanto, recentemente, algumas iniciativas e projetos que buscam ensinar ML para alunos surgiram principalmente como unidades extracurriculares (MARQUES et al., 2020). Essas unidades instrucionais ensinam competências que vão de apresentações sobre o que é ML a técnicas específicas da área, com ênfase em redes neurais artificiais e impactos do ML até a aplicação e criação de modelos de

Deep Learning. Considerando a complexidade, várias unidades instrucionais abordam apenas os processos mais simples, como gerenciamento de dados. Outras abrangem o processo completo de ML de uma forma simplificada, apresentando em diferentes graus de intensidade alguns dos processos subjacentes do ML, como o treinamento de redes neurais (MARQUES et al., 2020). Para o desenvolvimento de modelos de ML dentro do contexto educacional são geralmente adotadas ferramentas visuais como Google Teachable Machine (GOOGLE, 2020a) ou soluções customizadas como, p.ex., o LearningML (RODRÍGUEZ-GARCÍA et al., 2020) ou PIC (TANG et al., 2019), que também permitem uma fácil implantação de ML. Essas soluções permitem implantar modelos de ML em ambientes de programação baseados em blocos, como Scratch ou App Inventor. Com um foco maior no Ensino Médio, também há alguns cursos como os oferecidos usando linguagens textuais de desenvolvimento, como Python, em Jupyter Notebooks (TECHGIRLZ, 2020), ambiente que permite a execução do código no navegador *web*. As unidades instrucionais variam também de unidades escolares e cursos no modo presencial até cursos e tutoriais on-line.

Como parte do processo de aprendizagem, é importante avaliar a aprendizagem dos alunos fornecendo *feedback* tanto para o aluno quanto para o professor (HATTIE, TIMPERLEY, 2007). No entanto, apesar de muitos esforços para abordar a avaliação da educação em computação no Ensino Fundamental e Médio, com foco no pensamento computacional, algoritmos e programação (TANG et al., 2019; LYE, KOH, 2014), a maioria das unidades instrucionais em ML atualmente não propõe soluções para a avaliação do aluno (MARQUES et al., 2020). Poucos cursos de ML incluem avaliações, que são tipicamente baseadas em questionários muito simples, enquanto avaliações baseadas em desempenho são basicamente inexistentes. Apenas Sakulkueakulsuk et al. (2018) propõe uma avaliação baseada no desempenho do modelo de ML criado pelos alunos , enquanto o *AI Family Challenge* (TECHNOVATION FAMILIES, 2020) e o *Exploring Computer Science* (2020) avaliam o resultado ou a apresentação dos alunos por meio de rubricas. No entanto, nenhuma informação adicional sobre seu projeto ou avaliação foi encontrada, deixando sua eficácia e validade questionáveis.

Portanto, esta pesquisa pretende definir sistematicamente um modelo para avaliar a aprendizagem de conceitos e práticas de ML com base em artefatos de ML sendo criados por alunos de Ensino Médio como resultado da fase "*Use*" do ciclo "*Use-Modify-Create*" (LYTLE et al., 2019). O modelo de avaliação proposto é automatizado no contexto do curso de *Machine Learning* no Ensino Médio voltado à aprendizagem da tarefa de classificação de imagens usando fastai (HOWARD; GUGGER, 2020) em Jupyter Notebooks no Google Colab (GOOGLE, 2020d). Também é desenvolvida a apresentação visual dos resultados no Jupyter Notebook, utilizando-se da biblioteca *ipywidgets* (IPYWIDGETS, 2021).

1.2 OBJETIVOS

Nas seções a seguir são descritos o objetivo geral e os objetivos específicos deste trabalho.

OBJETIVO GERAL

O objetivo geral deste trabalho é desenvolver e automatizar um modelo de avaliação da aprendizagem de ML voltado à classificação de imagens no Ensino Médio. O modelo visa avaliar a aprendizagem de conceitos de ML desenvolvidos em Jupyter Notebooks no ambiente do Google Colab com Python no contexto do ensino de computação, focando principalmente no Ensino Médio.

OBJETIVOS ESPECÍFICOS

Os objetivos específicos do presente trabalho são:

Objetivo I – Síntese da fundamentação teórica em relação ao conceito de ensino de ML no Ensino Médio, *Machine Learning* com Jupyter e Python e avaliação de aprendizagem.

Objetivo II – Análise do estado da arte em relação a modelos de avaliação da aprendizagem de ML no contexto do Ensino Médio.

Objetivo III – Desenvolvimento do modelo conceitual para avaliar a aprendizagem de ML no Ensino Médio.

Objetivo IV – Desenvolvimento da automação do modelo de avaliação.

Objetivo V – Avaliação preliminar da qualidade da automação desenvolvida.

Delimitação do escopo

Este trabalho é focado somente na tarefa de classificação de imagens instanciado para a classificação de 6 espécies de árvores no contexto de um curso de *Machine Learning* voltado a classificação de imagens de árvores nativas de Santa Catarina (CARDOZO, 2021), especificamente desenvolvido utilizando o *framework* fastai em Jupyter Notebook executado no Google Colab. O foco do presente trabalho também é a avaliação com base no desempenho, não incluindo a automação do plano de avaliação de *quizzes* como parte do material interativo do curso.

1.3 METODOLOGIA DE PESQUISA

Nessa pesquisa é usada uma abordagem multi-método. A metodologia de pesquisa utilizada neste trabalho é dividida nas etapas apresentadas a seguir.

Etapa 1 – Fundamentação teórica

Atividade focada em estudar, analisar e sintetizar os conceitos principais e a teoria referente aos temas a serem abordados neste trabalho. Nesta etapa é apresentada a fundamentação teórica utilizando a metodologia de revisão narrativa (CORDEIRO et al., 2007) e são realizadas nas atividades abaixo.

Atividade 1.1 – Síntese sobre ensino de *Machine Learning* no Ensino Médio.
Atividade 1.2 – Síntese de conceitos do *Machine Learning* com Jupyter Notebook.
Atividade 1.3 – Síntese de avaliação de aprendizagem.

Etapa 2 – Estado da arte

Nesta etapa é realizado um mapeamento sistemático da literatura seguindo o processo proposto por (PETERSEN et al., 2008) para identificar e analisar modelos de avaliação da aprendizagem de ML no Ensino Médio. Esta etapa é dividida nas seguintes atividades:

Atividade 2.1 – Definição do protocolo de revisão.
Atividade 2.2 – Execução da busca e seleção de artigos relevantes.
Atividade 2.3 – Extração e análise de informações relevantes.
Atividade 2.4 – Discussão das informações encontradas.

Etapa 3 – Desenvolvimento do modelo de avaliação

Para o desenvolvimento do modelo de avaliação da aprendizagem de ML no Ensino Médio é seguida a metodologia de *design* instrucional ADDIE (Branch, 2009) para o desenvolvimento do curso como todo, usando para o desenvolvimento da avaliação o *Evidence-Centered Design* (MISLEVY et al., 2003). O desenvolvimento é dividido em três etapas. Análise é a primeira etapa que consiste na análise do contexto do público alvo. Em seguida, na etapa Projeto é definido o desempenho que deseja ser alcançado definindo um plano de avaliação. O objetivo da terceira etapa de Desenvolvimento é de gerar e/ou desenvolver todos os materiais didáticos: neste caso, especificamente os materiais de avaliação (rubrica).

Atividade 3.1 – Analisar e modelar o domínio.
Atividade 3.2 – Desenvolver o *framework* conceitual.
Atividade 3.3 – Projetar a implementação da avaliação.

Etapa 4 – Desenvolvimento da automação

Nesta etapa é desenvolvido a automação em várias iterações para cada um dos critérios definidos. É utilizado um processo iterativo/incremental de desenvolvimento de software (LARMAN; BASILI, 2003). A partir da análise de requisitos com base no modelo de avaliação, são realizadas várias iterações para automatizar a avaliação de forma incremental. Cada iteração envolve modelagem, implementação e teste de software.

Atividade 4.1 – Análise de requisitos.
Atividade 4.2 – Definição da arquitetura.
Atividade 4.3 – Iteração 1 - Avaliação.
Atividade 4.4 – Iteração 2 - Documentação com JSON.
Atividade 4.5 – Iteração 3 - Documentação com PDF.
Atividade 4.6 – Iteração 4 - Apresentação dos resultados.
Atividade 4.7 – Iteração 4 - Disponibilização da automação.

Etapa 5 – Avaliação preliminar da automação desenvolvida

Nesta etapa é realizada uma avaliação com o objetivo de analisar a qualidade da automação desenvolvida por meio de um painel de especialistas. A partir do objetivo da avaliação são derivados os fatores de qualidade e medidas a serem avaliadas seguindo GQM (BASILI et al., 1994). São coletados os dados por meio de um questionário realizando um teste de usabilidade. Esses dados coletados são analisados, interpretados e discutidos.

Atividade 5.1 – Definir a avaliação por meio de teste de usuários.
Atividade 5.2 – Executar o teste e coletar os dados dos testes de usuários.
Atividade 5.3 – Analisar e interpretar os dados coletados.

1.4 ESTRUTURA DO DOCUMENTO

No capítulo 2 é apresentada a fundamentação teórica sobre os conceitos que

sustentam a proposta do trabalho: o ensino de *Machine Learning* no Ensino Médio, ML com Jupyter Notebooks e avaliação de aprendizagem. No capítulo 3 o estado da arte é avaliado por meio de um mapeamento sistemático da literatura presente de modelos de avaliação da aprendizagem de ML no Ensino Médio. O capítulo 4 apresenta uma proposta de solução referente ao desenvolvimento do modelo da aprendizagem de ML no Ensino Médio e a rubrica resultante. O capítulo 5 relata o desenvolvimento da automação do modelo de avaliação proposto. O capítulo 6 apresenta os resultados da avaliação do modelo com um painel de especialistas e o capítulo 7 apresenta a conclusão e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ENSINO DE *MACHINE LEARNING* NO ENSINO MÉDIO

O *Machine Learning* é uma disciplina que estuda o uso de algoritmos e modelos que simulam atividades de aprendizagem humana e se autoaperfeiçoam, obtendo novos conhecimentos e habilidades (WANG et al. 2009). É uma importante área de estudo sobre aplicações de Inteligência Artificial, área ampla de ciência e engenharia que desenvolve e estuda técnicas para fazer com que computadores desempenhem tarefas de maneiras análogas à capacidade da inteligência humana. A competência de *Machine Learning* pode ser considerada essencial (DENNING, 2019), justificando a inclusão do ensino do conteúdo como parte da computação na Educação Básica.

A importância da Inteligência Artificial na sociedade, indústria e ciência torna-se cada vez mais evidente. Alguns países estão desenvolvendo iniciativas para suprir a necessidade de instruir seus estudantes, como exemplo a China, que tornou obrigatório que os alunos do Ensino Médio tenham Inteligência Artificial incorporada em seus currículos (JING, 2018). Deve-se assim inserir o conteúdo de IA e ML também no Ensino Médio, preparando alunos para uma tecnologia que já é força transformadora na sociedade. Dando esta oportunidade dos estudantes aprenderem sobre IA e ML no Ensino Médio espera-se que além de popularizar esta competência, motive-os também a procurar carreiras nesta área.

Iniciativas internacionais de diretrizes curriculares, como o *CSTA K-12 Computer Science Framework* (CSTA, 2016), buscam alcançar uma padronização e fornecer embasamento sobre quais conteúdos de computação os estudantes do Ensino Fundamental e Médio devem ser apresentados. Entretanto, o ensino de Inteligência Artificial e especificamente *Machine Learning* para esses estudantes não é bem definido. O *CSTA K-12 Computer Science Standards* (CSTA, 2017), por exemplo, não cita o ensino de *Machine Learning*, embora define que o estudante deve entrar em contato com Inteligência Artificial no Ensino Médio e com dois objetivos de aprendizagem:

- Descrever como a Inteligência Artificial move diversos *softwares* e sistemas físicos;
- Implementar um algoritmo de Inteligência Artificial que jogue um jogo contra um oponente humano ou resolva um problema.

Embora programas de ensino de Inteligência Artificial para o Ensino Médio ainda sejam raros (FORBES, 2019), empresas como Google, General Motors e NVIDIA vêm buscando iniciativas para ampliar a educação em IA, apoiando principalmente com investimentos (AI4ALL, 2018; ISTE, 2018) e programas de ensino (NVIDIA, 2018). Um projeto nesse contexto é o AI4K12 (AI4K12, 2020a), esforço conjunto das associações *Association for the Advancement of Artificial Intelligence (AAAI)* e *Computer Science Teachers Association (CSTA)*, com propósito principal de desenvolver guias para o ensino de Inteligência Artificial a estudantes de Ensino Fundamental e Médio (TOURETZKY et al., 2019b), além de centralizar recursos para professores e promover uma comunidade desenvolvedora destes recursos. Para estruturar as diretrizes referente ao ensino de IA na K-12, a iniciativa define 5 grandes ideias apresentadas na Tabela 1.

Tabela 1 - 5 grandes ideias em Inteligência Artificial (TOURETZKY et al., 2019b)

Ideia		**Descrição**
1	*Percepção*	Computadores percebem o mundo usando sensores
2	*Representação e raciocínio*	Agentes mantém modelos e representações do mundo e os usa para raciocinar
3	*Aprendizagem*	Computadores podem aprender a partir de dados
4	*Interação natural*	Agentes inteligentes necessitam ter diversos tipos de conhecimento para interagir com humanos de forma "natural"
5	*Impacto social*	A Inteligência Artificial pode impactar a sociedade de formas positivas e negativas

A grande ideia 3 (Aprendizagem) refere-se ao *Machine Learning,* indicando que os estudantes devem entender que ML é um tipo de inferência estatística utilizada para encontrar padrões nos dados. De acordo com a diretriz, os conceitos

fundamentais do assunto a serem ensinados aos estudantes durante o Ensino Fundamental e Médio devem incluir:

- O que é aprendizagem;
- Abordagens ao *Machine Learning* (algoritmos de regressão, algoritmos baseados em instância; máquinas de vetores de suporte, algoritmos de árvore de decisão, bayesianos, de *clustering*, de redes neurais artificiais, etc.);
- Tipos de algoritmos de aprendizagem, divididos por tipo de aprendizagem;
- Fundamentos de redes neurais;
- Tipos de arquiteturas de redes neurais;
- Como os dados de treinamento influenciam a aprendizagem;
- Limitações do *Machine Learning*.

A Tabela 2 apresenta um resumo do detalhamento da grande ideia 3 com seus respectivos objetivos de aprendizagem para cada nível de ensino.

Tabela 2 - Resumo dos objetivos de aprendizagem da grande ideia 3 (traduzido de (AI4K12, 2020b))

Conceito		*Anos escolares*			
		Ensino Infantil	**Ensino Fundamental - Anos Iniciais**	**Ensino Fundamental - Anos Finais**	**Ensino Médio**
Natureza de aprendizagem	Humanos vs. máquinas	Descrever e fornecer exemplos de como pessoas e computadores aprendem	Diferenciar entre como pessoas e computadores aprendem	Contrastar as características únicas do aprendizado humano e das maneiras que máquinas operam	Definir aprendizagem supervisionada, não supervisionada e por reforço e dar exemplos de aprendizagem humana similares a cada algoritmo
	Achando padrões em dados	Identificar padrões em dados rotulados e determinar as *features* que predizem os *labels*	Modelar como aprendizagem supervisionada identifica padrões em dados rotulados	Modelar como aprendizagem não supervisionada identifica padrões em dados não rotulados	Modelar como ML constrói um raciocinador para classificação ou predição ajustando os parâmetros do raciocinador (suas interpretações internas)
	Treinando um modelo	Demonstrar como treinar um computador para identificar algo	Treinar um modelo de classificação usando *Machine Learning*, e então examinar a precisão do mesmo com entradas novas	Treinar e avaliar um modelo de classificação ou predição usando *Machine Learning* em um conjunto de dados tabular	Usar um algoritmo de aprendizagem supervisionada ou não supervisionada para treinar um modelo com dados reais e avaliar os resultados
	Construir vs. usar um raciocinador	N/A	Demonstrar como dados de treino são rotulados quando se	Explicar a diferença entre treinar e usar um modelo de	Ilustrar o que acontece em cada etapa necessária ao

			usa uma ferramenta de ML	raciocínio	usar ML para construir um classificador ou preditor
	Ajustar representações internas	N/A	Analisar um jogo em que se constrói a árvore de decisão, descrevendo a organização da árvore e do algoritmo usado para adicionar nós	Comparar como um algoritmo de aprendizado de árvore de decisão funciona vs. como um algoritmo de aprendizado de redes neurais funciona	Descrever como vários tipos de algoritmos de ML aprendem ao ajustar suas interpretações internas
	Aprendendo por experiência	N/A	Explicar como aprendizado de reenforço permite que um computador aprenda por experiência (tentativa)	Explicar a diferença entre aprendizado supervisionado e não supervisionado	Selecionar o tipo de algoritmo de ML apropriado (aprendizado supervisionado, não supervisionado ou por reforço) para resolver um problema de raciocínio
Redes neurais	Estrutura de uma rede neural	N/A	Ilustrar como uma rede neural de 1 a 3 neurônios é uma função que computa uma saída	Ilustrar a estrutura de uma rede neural e descrever como suas partes formam um conjunto de funções que computam uma saída	Descrever as arquiteturas e usos das redes neurais *feedforward*, redes convolucionais 2D, recorrentes e adversariais generativas
	Ajuste de peso	N/A	Demonstrar como pesos são designados em uma rede neural para produzir o comportamento desejado de entrada e saída	Demonstrar como uma regra de aprendizado pode ser usada para ajustar os pesos em uma rede neural de um nível	Treinar uma rede neural multicamadas usando o algoritmo de aprendizagem de retropropagação e descrever como os pesos dos neurônios e as saídas das unidades ocultas mudam como resultado da aprendizagem
Conjuntos de dados	Conjuntos de *feature*	Criar um conjunto de dados rotulado com *features* explícitas para ilustrar como computador conseguem aprender a classificar coisas como alimentos, filmes ou brinquedos	Criar um conjunto de dados rotulado com *features* explícitas de diferentes tipos e usa uma ferramenta de *Machine Learning* para treinar um classificador nestes dados	Criar um conjunto de dados para treinar um classificador de árvore de decisão ou preditor e explorar o impacto que diferentes características têm na árvore de decisão	Comparar dois conjuntos de dados com dados reais em termos de *features* inclusas e como estas *features* estão codificadas
	Conjuntos de dados grandes	N/A	Ilustrar como treinar um classificador para um conceito amplo como “cachorro” requer uma grande quantidade de dados para capturar a diversidade do domínio	Ilustrar como objetos em uma imagem podem ser segmentados e rotulados para construir um conjunto de treino para reconhecimento de objetos	Avaliar um conjunto de dados usado para treinar um sistema de IA real considerando seu tamanho, a forma com que os dados foram adquiridos e rotulados, o armazenamento necessário, e o tempo estimado para produzi-lo
	Viés	Examinar um conjunto de dados rotulado e identificar problemas nos dados que poderiam levar o computador a fazer	Examinar características e rótulos de dados de treino para detectar possíveis fontes de viés	Explicar como a escolha dos dados de treino molda o comportamento do classificador, e como esse viés pode ser	Investigar desequilíbrios nos dados de treinamento em termos de gênero, idade, etnia e outras

		predições incorretas		introduzido se o conjunto de treino não for balanceado apropriadamente	variáveis demográficas que podem resultar em um modelo enviesado, utilizando alguma ferramenta de visualização

É esperado que os estudantes do Ensino Médio estejam preparados para treinar uma rede neural usando uma ferramenta interativa como o TensorFlow Playground (TENSORFLOW, 2020) e estudantes avançados estejam aptos a desenvolver aplicações simples de *Machine Learning,* usando ferramentas como scikit-learn (SCIKIT-LEARN, 2020) e Python (TOURETZKY, 2019b).

Há propostas de definição do conceito de alfabetização em Inteligência Artificial (LONG; MAGERKO, 2020), sendo um conjunto de competências que permitem que indivíduos avaliem de forma crítica as tecnologias de Inteligência Artificial. Espera-se que a pessoa alfabetizada comunique-se e colabore de forma efetiva com IA, utilizando-a como uma ferramenta *online*, em casa e em seu trabalho. A Tabela 3 apresenta as competências-chave definidas pelo trabalho para alcançar a alfabetização em IA, sendo as competências 9-13 especificamente relacionadas à ML.

Tabela 3 - Competências da alfabetização em IA. Traduzido de (LONG; MAGERKO, 2020)

#	Competência	Descrição
1	Reconhecendo IA	Distinguir entre artefatos tecnológicos que usam ou não IA
2	Entendendo inteligência	Analisar e discutir criticamente características que fazem uma entidade “inteligente”, incluindo discutir diferenças entre humanos, animais e inteligência de máquina
3	Interdisciplinaridade	Reconhecer que existem muitos jeitos diferentes de pensar e desenvolver máquinas “inteligentes”. Identificar uma variedade de tecnologias que usam IA, incluindo sistemas cognitivos, robótica e ML
4	Generalista vs. específico	Diferenciar entre IA generalista e IA específica
5	Pontos fortes e fracos de IA	Identificar tipos de problemas que IA se sobressai e problemas que são mais difíceis para a mesma. Usar essa informação para determinar quando é apropriado usar IA e quando usar habilidades humanas
6	Imaginar IA do futuro	Imaginar possíveis aplicações futuras de IA e considerar os efeitos de tal aplicação no mundo

7	Representação	Entender o que representação de conhecimento é e descrever alguns exemplos de representações de conhecimento
8	Tomada de decisão	Reconhecer e descrever exemplos de como computadores raciocinam e fazem decisões
9	Passos de ML	Entender as etapas envolvidas no *Machine Learning* e suas práticas e desafios
10	Papel humano na IA	Reconhecer que humanos possuem um papel importante no desenvolvimento, escolha de modelos e ajustes nos sistemas de IA
11	*Data literacy*	Compreender conceitos básicos de alfabetização de dados, como acessar, manipular, resumir e apresentar dados (PRADO; MARZAL, 2013)
12	Aprender pelos dados	Reconhecer que computadores geralmente aprendem a partir de dados, incluindo os que ele mesmo gera
13	Interpretar criticamente os dados	Entender que os dados não devem ser considerados pelo seu valor bruto e requerem interpretação; Descrever como os conjuntos de treinamento podem afetar os resultados de um algoritmo
14	Ação e reação	Entender que alguns sistemas de IA têm a habilidade de fisicamente agir no mundo. Essa ação pode ser direcionada por raciocínio de alto nível (como andar em um caminho planejado) ou pode ser reativo (pular para trás para evitar um obstáculo percebido)
15	Sensores	Entender o que sensores são, reconhecer que computadores percebem o mundo usando sensores, e identificá-los em uma variedade de dispositivos. Reconhecer que diferentes sensores suportam diferentes tipos de representação e raciocínio sobre o mundo
16	Ética	Identificar e descrever diferentes perspectivas no problemas éticos chave acerca de IA (privacidade, singularidade, tomadas de decisões éticas, etc.)
17	Programabilidade	Entender que os agentes são programáveis

No Brasil o currículo do Ensino Médio é composto pela Base Nacional Comum Curricular (BNCC), documento normativo que define o conjunto de aprendizagens essenciais que os alunos devem desenvolver ao longo das etapas e modalidades da Educação Básica. De forma complementar às quatro áreas principais (Linguagens e suas Tecnologias, Matemática e suas Tecnologias, Ciências da Natureza e suas Tecnologias, Ciências Humanas e Sociais Aplicadas) na composição do currículo, estão os itinerários formativos (MINISTÉRIO DA EDUCAÇÃO, 2017): conjuntos de unidades curriculares ofertadas pelas escolas e redes de ensino, possibilitando ao estudante aprofundar seus conhecimentos em uma área. A BNCC cita Inteligência Artificial como possível tema de itinerário

formativo integrado (que mobiliza competências e habilidades de diferentes áreas) sob a área de Matemática e suas Tecnologias; também cita a importância dos algoritmos associados ao pensamento computacional, tratando-os novamente como possível objeto de estudo.

> Os itinerários formativos – estratégicos para a flexibilização da organização curricular do Ensino Médio, pois possibilitam opções de escolha aos estudantes – podem ser estruturados com foco em uma área do conhecimento, na formação técnica e profissional ou, também, na mobilização de competências e habilidades de diferentes áreas, compondo itinerários integrados, nos seguintes termos das DCNEM/2018:
> [...] II – matemática e suas tecnologias: aprofundamento de conhecimentos estruturantes para aplicação de diferentes conceitos matemáticos em contextos sociais e de trabalho, estruturando arranjos curriculares que permitam estudos em resolução de problemas e análises complexas, funcionais e não-lineares, análise de dados estatísticos e probabilidade, geometria e topologia, robótica, automação, **inteligência artificial**, programação, jogos digitais, sistemas dinâmicos, dentre outros, considerando o contexto local e as possibilidades de oferta pelos sistemas de ensino; (MINISTÉRIO DA EDUCAÇÃO, 2017)

A Sociedade Brasileira de Computação manifestou sua discordância e inseriu diversas críticas em uma análise técnica dos pontos citados sobre computação na versão homologada da BNCC (SBC, 2018b), como a falta de ensino de construção de algoritmos, linguagem inadequada e habilidades mal formuladas. A própria SBC também desenvolveu uma proposta de itinerário formativo de computação (SBC, 2018a) dividida em três eixos principais: Cultura Digital, Pensamento Computacional e Mundo Digital. No eixo Mundo Digital, espera-se que o aluno do Ensino Médio tenha a compreensão de fundamentos sobre Inteligência Artificial, conforme apresentado na Tabela 4. Porém, pouco destaque é dado para IA e *Machine Learning* nem é citado nas diretrizes.

Tabela 4 - Objetivo de conhecimento sobre IA (SBC, 2018a)

Computação: Ensino Médio	
Inteligência artificial e robótica	Compreender os fundamentos da inteligência artificial e da robótica

Para efetivamente transmitir a aprendizagem de *Machine Learning* aos estudantes, usualmente utiliza-se de unidades instrucionais (UIs). As unidades instrucionais são compostas por um conjunto de aulas (em formato de curso,

workshop, etc.) com o foco em ensinar determinados objetivos de aprendizagem para um público-alvo. Para permitir que os estudantes construam modelos mentais corretos e aprendam competências no nível de aplicação, é importante adotar metodologias ativas envolvendo ativamente os alunos no processo de aprendizagem.

Observa-se que nos últimos anos foram desenvolvidas diversas unidades instrucionais voltados ao ensino de IA e ML no Ensino Médio (MARQUES et al., 2020), em que as abordagens ativas de aprendizagem predominam fortemente, complementadas com atividades como vídeos e apresentações. As unidades variam muito em termos de níveis de aprendizagem que os alunos devem adquirir de acordo com a taxonomia de Bloom (BLOOM et al., 1956). A maioria das unidades instrucionais se concentram com exclusividade nos níveis de aprendizagem mais baixos (conhecimento e compreensão). Algumas UIs também abordam o nível de aplicação, fazendo com que os alunos criem seus próprios modelos de ML.

As unidades instrucionais tipicamente adotam o ciclo "*Use-Modify-Create*" (LYTLE et al., 2019). Usado como técnica para ensino de computação, funciona de forma progressiva em que o aluno: (1) usa determinados artefatos, como modelos de *Machine Learning*, por meio de *interfaces* simples e com retorno rápido para entender sobre treino, teste e importância dos dados, (2) modifica artefatos existentes, alterando hiperparâmetros e utilizando diferentes conjuntos de dados, (3) eventualmente cria seus próprios artefatos, resolvendo um problema de ML e criando sua própria aplicação (KONG et al., 2020).

Os conteúdos abordados nas UIs voltadas ao Ensino Médio costumam ser visão computacional, reconhecimento facial, classificação de objetos, jogos e reconhecimento de voz (MARQUES et al., 2020). A maioria das unidades são fornecidas no formato de oficinas ou cursos presenciais, geralmente voltados a iniciantes e com tempo de duração curto. Os processos de *Machine Learning* abordados são gerenciamento de dados, aprendizagem do modelo, avaliação do modelo e *deployment*. Alguns dos ambientes utilizados nas UIs são baseados em blocos (Scratch e App Inventor) e outros em texto como Jupyter Notebook executado no ambiente do Google Colab utilizando Python. Adicionalmente, embora quase

todas UIs são em língua inglesa, há iniciativas em língua portuguesa, como o *Machine Learning* Para Todos (GRESSE VON WANGENHEIM et al., 2020) e Introdução a *Machine Learning* (ANDRADE et al., 2020).

2.2 *MACHINE LEARNING* COM JUPYTER NOTEBOOK

2.2.1 *MACHINE LEARNING*

Machine Learning é um campo de estudo da Inteligência Artificial que se refere ao desenvolvimento de algoritmos capazes de aprender com a experiência e adaptar-se sem serem explicitamente programados (SAMUEL, 2000), a partir de um conjunto de dados e de computações realizadas sobre estes. Utilizando diferentes tipos de algoritmos e modelos estatísticos, gera-se novas formas de conhecimento sobre o domínio do problema em questão. A aprendizagem gerada pode auxiliar na tomada de decisão humana e servir de entrada para o próprio algoritmo em um processo de *feedback* ou para outros modelos, agentes e sistemas. Esse tipo de aprendizagem é útil para o desenvolvimento de *softwares* que visam solucionar problemas em áreas que o conhecimento humano possui pouca *expertise*, problemas que dependam de um conjunto de fatores que variam com o tempo e ambiente (ALPAYDIN, 2010) ou que não sejam possíveis de ser resolvidos à maneira tradicional, seguindo uma sequência linear de instruções.

A abordagem atual mais recente de *Machine Learning,* muito eficiente em termos de tempo de desenvolvimento e custo para alguns problemas, é o *Deep Learning* (DARGAN et al., 2020). O DL tem tido papel crucial em avanços significativos e ganho de desempenho em diversas aplicações, como classificação de imagens, visão computacional, detecção de objetos, cidades inteligentes, etc. DL não está restrito à uma abordagem de aprendizado, mas possui diversos procedimentos e topografias que podem ser aplicados a um espectro variado de problemas complexos (DARGAN et al., 2020). Enquanto o ML necessita de algoritmos para interpretar os dados, aprender a partir deles e então sintetizar decisões com base em seu aprendizado, DL constrói algoritmos multicamadas em

um sistema de redes neurais artificiais, capazes de aprender e tomar decisões por conta própria. Redes neurais são unidades de processamento que conseguem adquirir conhecimento do ambiente por um processo de aprendizagem e guardá-lo em suas conexões (GUERESEN; KAYAKUTLU, 2011). As redes neurais são inspiradas nos neurônios biológicos, simulando suas conexões, pesos sinápticos e respostas a impulsos.

A saída do processo de aplicação de um algoritmo de aprendizagem em um conjunto de dados é denominado modelo. Os modelos gerados por *Machine Learning* procuram melhorar sua aprendizagem e sua acurácia na resolução de problemas conforme vão processando mais dados (DEEPAI, 2019). Esse processo de melhoria do modelo aplicado aos dados é chamado de treinamento. O modelo treinado é capaz então de realizar previsões de como as características de um exemplo irão se comportar em determinado cenário, inclusive para valores que não estavam presentes em seus dados de treinamento. Se as previsões forem de valores contínuos, o modelo é de regressão; se forem valores discretos, de classificação.

A aprendizagem depende do conjunto de dados (*dataset*) fornecidos ao algoritmo como dados de treinamento. Sendo assim, o processo de coletar um grande volume de dados representativos e relevantes é essencial para o *Machine Learning*. É justificável então afirmar que parte dos avanços recentes na área pode ser atribuída à capacidade atual de armazenamento, processamento e transmissão de dados (BAŞTANLAR; ÖZUYSAL, 2013). Uma forma de classificar diferentes tipos de aprendizagem que o *Machine Learning* pode obter é avaliando a quantidade de informações que os dados de treinamento possuem. Os dados podem ser de diversos tipos, como imagens, áudios, vídeos, etc. Diversas aplicações atuais usam imagens como entrada de dados, como aplicações de classificação de imagens. Imagens são dados altamente dimensionais, o que usualmente implica na necessidade de um maior volume de dados para efetuar o treinamento do modelo.

Há na literatura a noção de que 1000 imagens representativas para cada classe a ser classificada devem ser utilizadas (WARDEN, 2017). Este número surge do modelo vencedor do desafio original da rede ImageNet (RUSSAKOVSKY et al.,

2015). Entretanto, é possível obter bons resultados com poucas amostras utilizando-se de redes neurais convolucionais ou usando *Transfer Learning* com modelos pré-treinados (CHOLLET, 2016). No *Transfer Learning* o modelo é pré-treinado com imagens, preferencialmente do mesmo domínio do problema (ex. árvores, carros, animais), sendo possível obter resultados relevantes adicionando poucas imagens ao modelo. Uma maneira de melhorar o desempenho do *Transfer Learning* é aplicando o processo de *Fine-Tuning*, que ajusta o modelo pré-treinado ao organizar as camadas da rede e treina-o novamente para executar uma tarefa similar (GUO et al., 2019).

Outro aspecto a ser levado em consideração é a representatividade e qualidade dos dados utilizados no treinamento. O ideal é que os dados sejam o mais próximo possível das entradas que o modelo irá classificar. Em uma tarefa de reconhecimento de imagens, por exemplo, é desejável que sejam tiradas no mesmo ambiente das entradas a se classificar (WARDEN, 2017). As imagens devem conter poucos objetos/somente o objeto a ser classificado, não estarem desfocadas ou borradas, etc. Além disso, a quantidade de imagens para cada classe deve ser equilibrada para evitar um enviesamento da classificação do modelo.

Existem diferentes formas de aprendizagem, entre elas aprendizagem supervisionada e não supervisionada. A aprendizagem não supervisionada recebe como entrada somente os artefatos (p. ex. imagens). Assim, o algoritmo precisa identificar padrões, similaridades e diferenças que existem nos dados sem um treinamento prévio (SODHI et al., 2019). Um exemplo de uso da aprendizagem não supervisionado é nas técnicas de *clustering*, que categorizam dados de acordo com padrões e características em comum, como no algoritmo *k-means.*

Por outro lado, na aprendizagem supervisionada os dados do conjunto possuem etiquetas (*labels)*, que indicam a categoria de cada uma dos artefatos de entrada, como as imagens. Exemplos de aprendizagem supervisionada são algoritmos de classificação e regressão (BAŞTANLAR; ÖZUYSAL, 2013).

Ao final do treinamento do modelo é avaliado o desempenho do modelo medindo a diferença entre os valores previstos e os valores esperados, como no

caso da aprendizagem supervisionada. É importante que o erro da avaliação de desempenho seja mensurado em um conjunto de dados diferente do que foi utilizado para o treinamento, para garantir que o modelo está fazendo predições corretamente e de forma generalizada. Uma estratégia comum é dividir o conjunto de dados disponível para o treinamento do modelo em conjuntos de treinamento e de validação, seguindo tipicamente uma proporção de 70-80% e 20-30% (AWS, 2020a), respectivamente. As medidas de desempenho dependem do tipo de aprendizagem como também da tarefa específica de ML.

Essa diferença entre os valores previstos e valores esperados na predição do modelo é considerada uma falha. Uma abordagem para avaliar a qualidade do modelo é calcular a razão do número de exemplos classificados corretamente pelo modelo pela quantidade total de exemplos fornecidos (NOVAKOVIĆ et al., 2017). Naturalmente, essa abordagem parte do princípio que todas as falhas são igualmente importantes, o que na prática dificilmente é o que acontece. Uma alternativa é apresentar os tipos de erros em uma matriz bidimensional conhecida como matriz de confusão. Nessa matriz, cada linha corresponde a uma classe contendo os valores que foram preditos e cada coluna possui o valor que deveria ter sido classificado corretamente. A matriz permite uma análise mais completa entre diferentes tipos de erros produzidos durante o treinamento (NOVAKOVIĆ et al., 2017).

Durante o treinamento tipicamente se avalia o desempenho da predição de um modelo em termos de conseguir prever a saída esperada analisando a perda (*loss*). A perda é uma penalidade para uma predição incorreta; um número que indica o quão ruim foi a predição para um único exemplo (GOOGLE, 2020b). Quanto mais distante de zero, pior é a predição; quando a predição acerta corretamente, a perda é zero. Na aprendizagem supervisionada, o processo de construção de modelos de *Machine Learning* visa encontrar um modelo com o mínimo de perda possível. Uma forma de agregar os erros individuais para cada exemplo de um conjunto de dados é com uma função de perda (*loss function).* Há diferentes tipos de funções de perda que se aplicam para algoritmos de classificação, como perda de

articulação (*hinge loss)* e para algoritmos de regressão, como perda quadrática (*mean square error*) e de entropia cruzada (*log loss*) (ALGORITHMIA, 2018).

Dado que quanto menor a perda do modelo, melhor seu desempenho, é possível otimizá-lo encontrando formas de minimizar a função de perda. Um algoritmo usado para atingir este objetivo é o gradiente descendente, que consiste em encontrar, de forma iterativa, os valores dos parâmetros que minimizam a função de perda. Por exemplo, a função de custo *C* para os parâmetros *v1* e *v2* pode ser representada da forma conforme apresentada na Figura 1.

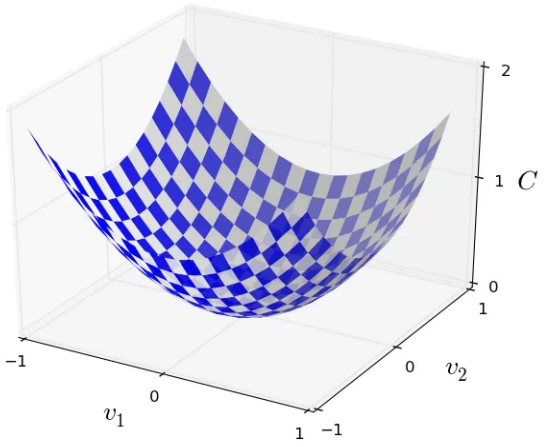


Figura 1 - Diagrama de representação do gradiente descendente (GUPTA, 2017)

O algoritmo então visa encontrar o ponto mais baixo dessa função, que no caso da regressão linear é convexa. As condições de realização das iterações são definidas por hiper parâmetros: parâmetros utilizados para controlar o processo de aprendizagem do modelo, informados ao modelo antes do início do treinamento. A taxa de aprendizagem (*learning rate*) define o tamanho dos passos que são dados em direção ao valor mínimo; se os passos forem pequenos, é necessário muitas iterações no treino, o que pode torná-lo lento, embora possivelmente o algoritmo encontrará o valor mínimo em algum momento. Se os passos forem muito grandes, é possível que o valor nunca seja encontrado. A quantidade de passagens completas pelo conjunto de treinamento por iteração é chamada de épocas (*epochs*), e o número de exemplos utilizados para o cálculo do gradiente em uma

única iteração é denominado *batch*, ou lote (GOOGLE, 2020c). A maioria dos problemas de *Machine Learning* requer refinamento nos hiperparâmetros do treinamento do modelo, e os valores para encontrar um erro mínimo de forma eficiente variam conforme o algoritmo, quantidade e qualidade dos dados de treinamento, etc., caracterizando o processo como experimental.

O desenvolvimento de modelos de *Machine Learning* de maneira eficiente e eficaz usualmente é feito seguindo processos e metodologias estabelecidas pela indústria e comunidade. Há diferentes fluxos de trabalhos recomendados, como os propostos por Microsoft (AMERSHI et al., 2019) e Google (GOOGLE CLOUD, 2018). Muito do que existe nestes processos foi baseado em metodologias da área de *Data Mining*, como KDD (FAYYAD et al., 1996) e CRISP-DM (WIRTH; HIPP, 2000). Embora existam pequenas diferenças entre como as fases são nomeadas e organizadas, os fluxos de trabalho mantêm em comum a essência de serem centrados nos dados e com múltiplos *feedback loops* entre as diferentes fases do processo.

Na fase de requisitos do modelo é definido quais são características dos dados serão utilizadas e quais são possíveis algoritmos ideais para o problema a ser resolvido (AMERSHI et al., 2019). Posteriormente, os dados são coletados, capturados por quem está construindo o modelo ou integrados de conjuntos de dados já existentes. Os dados precisam ser limpos, tendo valores discrepantes ou incorretamente preenchidos removidos (AMERSHI et al., 2019). Depois, são rotulados, quando dados brutos são identificados por meio de rótulos significativos e informativos, fornecendo contexto aos algoritmos de ML (AWS, 2020b).

Com os dados preparados, eles são adaptados para serem usados como entrada aos algoritmos previamente estabelecidos e rótulos relevantes são extraídos ou novos são criados durante a fase de *feature engineering*. Após preparar os dados e definir quais informações serão utilizadas, inicia-se efetivamente o treinamento do modelo utilizando algum algoritmo de ML.

O modelo pré treinado pode então ser compartilhado com a comunidade, como realizado em plataformas agregadoras de modelos como *Model Zoo.* Modelos tipicamente usados para a tarefa de classificação de imagens são redes neurais

residuais ou ResNets (HE et al., 2015) com diferentes camadas de profundidade, p.ex., ResNet-18 e ResNet-50. Para aplicativos móveis, usa-se modelos de redes neurais leves como MobileNets (HOWARD et al., 2017).

O resultado do treinamento é então avaliado utilizando novos conjuntos de dados e, caso não seja satisfatório, volta-se à fase de treinamento do modelo, efetuando nova análise dos dados e execução dos algoritmos (AMERSHI et al., 2019). Para a avaliação de modelos de classificação de imagens, geralmente analisa-se medidas referentes aos erros de predição do modelo, como acurácia (razão entre as previsões corretas e o número total de previsões nos dados de treinamento) total e por categoria, além de interpretações da matriz de confusão e *top losses*. Outras medidas utilizadas são precisão (razão de instâncias relevantes entre as instâncias recuperadas) e *recall* (razão de instâncias relevantes que foram recuperadas) (POWERS, 2020).

Finalmente, o modelo é implantado no ambiente desejado para fazer predições para imagens nunca vistas antes e monitorado para encontrar possíveis erros. Embora o processo como um todo siga um fluxo linear, há vários pequenos ciclos envolvidos e é natural que se refaça alguma etapa, visando maximizar o desempenho do modelo.

2.2.2 DESENVOLVIMENTO DE ML COM JUPYTER NOTEBOOK

As aplicações de *Machine Learning* requerem um grande volume de dados e capacidade de processamento, o que torna a questão de ensino ainda mais complexa: a configuração, hardware e suporte de TI necessário para replicar um ambiente pode torná-la inviável para uma aula com uma quantidade razoável de alunos. Uma ferramenta que soluciona essa questão é o Jupyter Notebook, antigo IPython Notebook, hoje um padrão na comunidade de cientistas de dados (PERKEL, 2018). De código aberto, pode ser executado no navegador *web* e funciona como um laboratório virtual para compartilhamento de *workflows*, códigos, dados e visualizações durante o processo de pesquisa. O Jupyter Notebook incorpora os princípios FAIR (*Findable*, *Accessible*, *Interoperable*, *Reusable*) para objetos digitais, que justifica sua utilidade como possível ferramenta para a comunicação científica

(RANDLES et al., 2017). É uma forma de computação interativa, dado que fornece um ambiente para os usuários executarem códigos, analisar o resultado, modificar os *inputs* e testar suas hipóteses de forma iterativa e acessível.

Jupyter Notebook suporta mais de 40 linguagens de programação, como Python, R e Julia, *outputs* interativos, integração com ferramentas de *big data* e um ecossistema de ambientes *on-line* de renderização e execução, como nbviewer (NBVIEWER, 2020) e Google Colaboratory (GOOGLE, 2020d). Também conhecido como Colab, este último é amplamente utilizado pela comunidade de *Machine Learning* para desenvolver e treinar redes neurais de forma *on-line*, divulgar pesquisas em Inteligência Artificial e criar tutoriais e manuais interativos.

Os *notebooks* são constituídos em sua forma básica por dois tipos de componentes no front-end, renderizados no navegador.

- Células de texto, usualmente com explicações e comentários na linguagem de marcação Markdown;

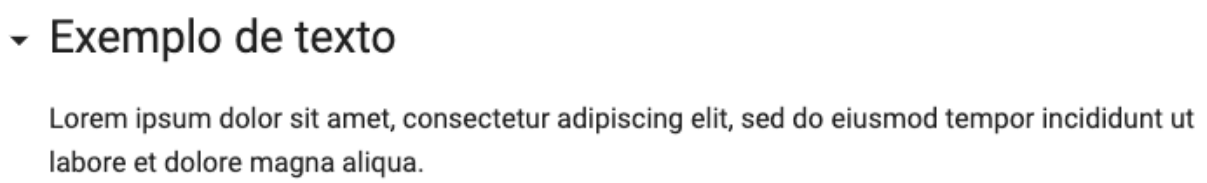


Figura 2 - Exemplo de célula de texto no Google Colab

- Células de código, com código fonte.

```python
import numpy as np

import tensorflow as tf
import tensorflow_hub as hub
import tensorflow_datasets as tfds

import matplotlib.pyplot as plt

print("Version: ", tf.__version__)
print("Eager mode: ", tf.executing_eagerly())
print("Hub version: ", hub.__version__)
print("GPU is", "available" if tf.config.list_physical_devices('GPU') else "NOT AVAILABLE")
```

```
Version:  2.3.0
Eager mode:  True
Hub version:  0.10.0
GPU is NOT AVAILABLE
```

Figura 3 - Exemplo de célula de código em Python utilizando TensorFlow, NumPy (NUMPY, 2020) e Matplotlib e output *de sua execução no Google Colab*

Também é possível visualizar diferentes tipos de multimídia, como os gráficos da Figura 4, *plots* realizados com a biblioteca matplotlib (MATPLOTLIB, 2020)

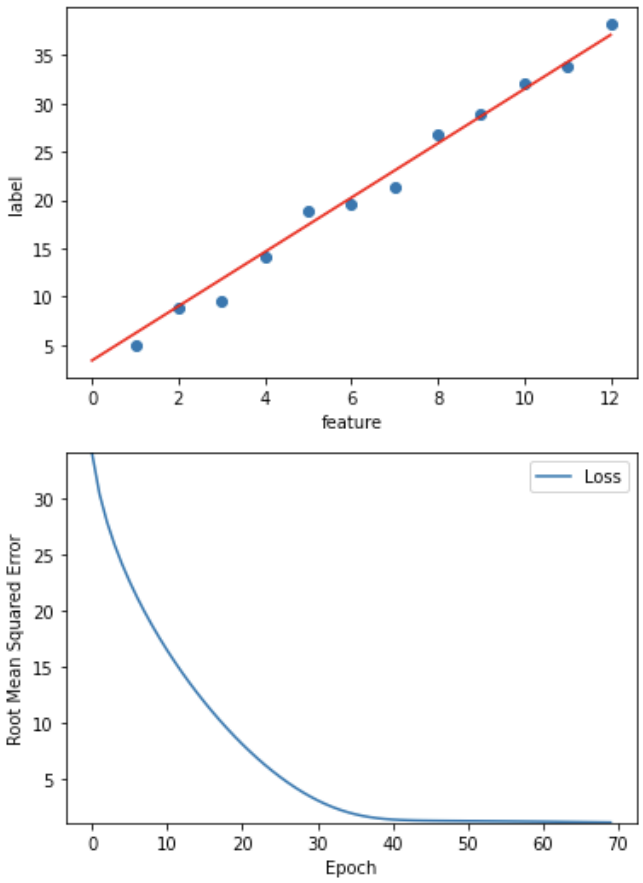


Figura 4 - Exemplo de plotagem de previsões de modelo regressão linear e curva de perda com matplotlib

Ao executar as células de código, o navegador encaminha as instruções para um *kernel* no *back-end*, responsável por executar o código e retornar os resultados. Os *kernels* podem ser executados diretamente no computador do usuário ou em um ambiente na nuvem, como no Google Colab. Em situações que requerem processamento de grande volume de dados, comum no treinamento de modelos de *Machine Learning*, a execução na máquina local é impraticável. Utilizando-se do Google Colab, é possível realizar o treinamento em um *back-end* executando em servidores distribuídos geograficamente e ter um *front-end* local.

2.3 AVALIAÇÃO DE APRENDIZAGEM

A avaliação é uma etapa importante no processo de aprendizagem do aluno. Sua função no processo de aprendizagem é servir para medir se o aluno sendo avaliado alcançou efetivamente os objetivos educacionais propostos, consequentemente verificando a qualidade do ensino realizado (TYLER, 1949, MARTINS; GUISSO, 2019). No processo de avaliação da aprendizagem também pode ser considerado as dimensões subjetivas e contextuais do aluno, extraindo suas principais competências conforme a possibilidade e necessidade, de forma que se apoie e se estimule a construção da aprendizagem (OLIVEIRA et al., 2016).

O planejamento de como a avaliação deve ser realizada decorre da definição das competências a serem desenvolvidas pelos alunos como resultado da aprendizagem. Competência é o resultado da mobilização de recursos por parte do indivíduo em três dimensões: (1) conhecimento, (2) habilidades e (3) atitudes para atuação em resposta a desafios (MORAES; RODRIGUES, 2019). A partir das competências a serem adquiridas com base nas diretrizes curriculares e contexto, os objetivos de aprendizagem são definidos.

A avaliação do ensino é um processo essencial tanto para o aluno quanto para o professor, indicando ao aluno como *feedback* quais competências que

precisa desenvolver ou se está desenvolvendo-as como desejado. Para o professor, serve de acompanhamento de sua efetividade na transmissão de conhecimento. Esse processo de *feedback* é constante e cíclico, conhecido como *feedback loop* apresentado na Figura 5.

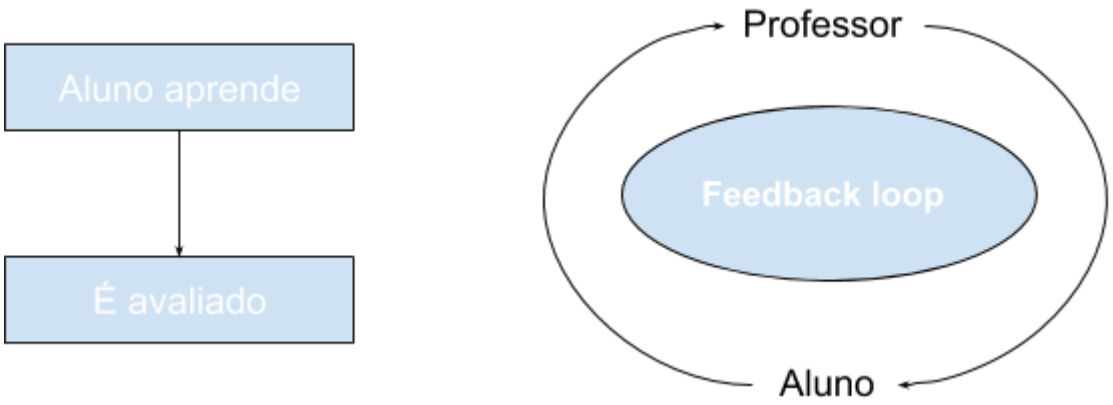
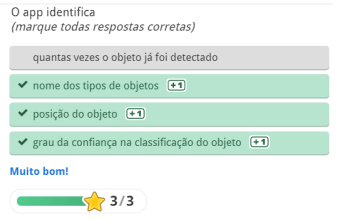
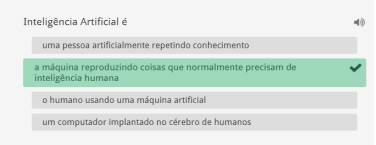
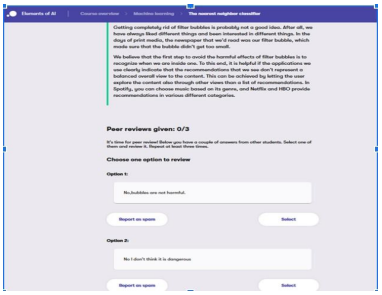
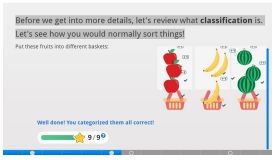
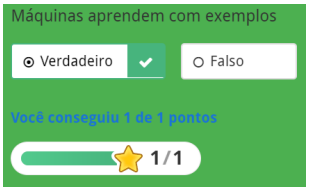


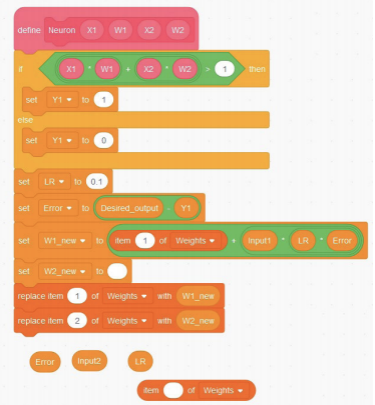
Figura 5 - Processo de avaliação e feedback loop *entre aluno e professor. Adaptado de (MONTALTI, 2016)*

TIPOS DE AVALIAÇÃO

A variedade das expectativas de aprendizagem implica em uma variedade de formas de avaliação, como *quizzes* com questões de múltipla escolha, folhas de exercício, avaliação por desempenho, entre outros. Os critérios utilizados para a escolha de técnicas e instrumentos dependem dos objetivos da avaliação, complexidade do conteúdo, tempo disponível, condições de ensino, além da quantidade, idade e perfil dos avaliados. Visando especificamente o modo de ensino não presencial, a Tabela 5 apresenta diversos tipos de avaliação.

Tabela 5 - Tipos de avaliação de ambientes virtuais de aprendizagem

Tipo de avaliação	Descrição	Exemplos em cursos de ML
Múltipla escolha	Permite criar questões flexíveis com uma lista de possíveis respostas, podendo ter apenas uma única resposta correta ou múltiplas respostas corretas. Avalia compreensão do aluno sobre fatos, ideias e aplicação de princípios (ELEARNING INDUSTRY, 2018)	 (GRESSE VON WANGENHEIM et al, 2020)  (GRESSE VON WANGENHEIM et al., 2020)
Redação	O aluno produz um pequeno texto sobre um assunto e é avaliado pela sua exposição de ideias e informações conforme o tema proposto. Varia de um parágrafo à páginas, é apropriada para avaliações que não podem ser realizadas com outros tipos de perguntas, como as que visam avaliar níveis mais altos de domínio do aluno (RAGUPATHI, 2020). Questões abertas se encaixam nessa categoria	 (ELEMENTS OF AI, 2019)
Arrastar e soltar	Permite criar tarefas de arrastar e soltar com imagens ou textos	 (READYAI, 2019)
Verdadeiro/falso	Permite criar questões de verdadeiro ou falso	 (GRESSE VON WANGENHEIM et al., 2020)

Avaliação de desempenho com base no artefato criado como resultado de atividade prática	Permite criar artefatos variados decorrentes de uma atividade prática, como código Scratch ou um modelo de ML	 (ESTEVES et al., 2019)

A avaliação por desempenho exige que o aluno crie soluções/artefatos, demonstrando na prática que adquiriu o conhecimento e consegue aplicá-lo de forma correta. A avaliação por desempenho é tipicamente feita por meio de rubricas (STEVENS; LEVI, 2005), utilizadas para comunicar ao estudante os critérios utilizados na correção e facilitar o fornecimento de *feedback* justo e oportuno por parte do instrutor (SEWELL et al., 2010). Além de fornecer elementos necessários para a avaliação, a rubrica também especifica o nível de desempenho necessário para alcançar os objetivos de aprendizagem esperados. A rubrica usualmente é composta por quatro componentes relacionados à tarefa: (1) descrição detalhada; (2) dimensões, referindo-se aos critérios que são avaliados; (3) escala que descreve diferentes níveis de desempenho; (4) descrição dos diferentes níveis de desempenho em cada uma das dimensões da tarefa.

Uma forma de realizar a avaliação é levando como base os níveis de aprendizagem da taxonomia de Bloom (BLOOM et al., 1956). A Taxonomia de Bloom pode servir de guia para selecionar avaliações de acordo com o(s) nível(is) de aprendizagem que se espera que os alunos atinjam. Conforme o contexto do presente trabalho, a Tabela 6 apresenta os respectivos tipos de avaliação para os três primeiros níveis de aprendizagem focados na Educação Básica.

Tabela 6 - Mapeamento dos níveis de aprendizagem e exemplos de atividades avaliativas. Adaptado de (CLICK4IT, 2013; RAGUPATHI, 2020; SEWELL; THEDE, 2009)

Nível	**Descrição**	**Tipos de avaliação**

Conhecimento	Lembrar ou reconhecer informações	Múltipla escolha, arrastar e soltar, verdadeiro/falso
Compreensão	Entender significados, interpretar, traduzir	Múltipla escolha, redação
Aplicação	Usar ou aplicar conhecimento, usar conhecimento em resposta a circunstâncias reais	Artefato criado como resultado de atividade prática, redação

As avaliações, de acordo com o *design* instrucional, podem ser classificadas de acordo com o momento em que são aplicadas. A avaliação é diagnóstica quando avalia o conhecimento do estudante antes do momento da atividade de aprendizagem acontecer. Já a avaliação formativa é realizada durante o ensino, de forma a revisar o processo de instrução para que este seja mais efetivo (ALVES et al., 2019). Encerra-se o processo de aprendizagem com a avaliação somativa, executada no final do curso para demonstrar o grau de atingimento dos objetivos de aprendizagem (BRANCH, 2009).

Em relação a aplicação da avaliação no contexto de ambientes virtuais de aprendizagem, em cursos *online* sem moderação dá-se por várias formas: automatizada, por pares, autoavaliação e pelo instrutor de forma manual. Na avaliação automatizada, os alunos obtêm *feedback* automático com seus acertos e erros em uma atividade. Outro tipo de avaliação é a avaliação por pares, em que os alunos executam uma atividade e são solicitados a avaliar a atividade de outro colega. Outro tipo de avaliação comum é a autoavaliação, em que os alunos são solicitados a avaliar seu próprio trabalho (PAPATHOMA, 2015). Quando o curso possui moderação, as atividades podem ser também avaliadas e/ou facilitadas por um instrutor.

CONFIABILIDADE E VALIDADE DA AVALIAÇÃO

Os métodos de avaliação precisam ser confiáveis e validados buscando a confiabilidade de avaliações entre diferentes educadores e/ou momentos de tempo. Além disso, é importante validar se os critérios que estão sendo avaliados continuam a medir a aprendizagem da competência em questão. A confiabilidade pode ser

definida como a extensão em que um instrumento de pesquisa produz resultados semelhantes em diferentes circunstâncias, assumindo que nada é alterado (ROBERTS; PRIEST, 2006). Ou seja, se o instrumento é considerado replicável (GOLAFSHANI, 2003). Já a validade determina se o instrumento realmente mede o que se pretende medir ou quão verdadeiros são os resultados do instrumento de avaliação (GOLAFSHANI, 2003).

Há diferentes formas de analisar a confiabilidade e a validade de um instrumento de avaliação. Em relação à confiabilidade, tipicamente é adotado o Alfa de Cronbach para analisar a consistência interna do instrumento. É calculado correlacionando a pontuação de cada item da escala (por exemplo, participantes de testes e questionários) com a pontuação total de cada observação. Em seguida, compara-se com a variância de todas as pontuações de itens individuais do instrumento (GOFORTH, 2015). O resultado é um coeficiente α expresso por um número entre 0 e 1 (TAVAKOL; DENNICK, 2011), em que o coeficiente 1 representa a máxima confiabilidade de consistência interna.

Em relação à análise da validade, tipicamente é feita uma análise de correlação referente à validade do construto. A análise é utilizada, por exemplo, para descobrir o quanto uma variável interfere no resultado de outra. Uma alta correlação é encontrada entre itens da mesma categoria e baixa correlação entre itens de categorias diferentes. O resultado da análise de correlação é um coeficiente de correlação cujos valores variam de -1 a +1. O coeficiente de correlação +1 indica que as duas variáveis estão perfeitamente relacionadas de maneira positiva e, -1, de maneira negativa. Já o coeficiente de correlação de zero indica que não há relação linear entre as variáveis avaliadas (GOGTAY; THATTE, 2017).

Complementando a análise de construto, tipicamente se analisa a decomposição do modelo de avaliação em dimensões/fatores e itens a serem medidos, pode ser também feito uma análise fatorial. É uma técnica estatística que analisa as relações entre um conjunto de itens de pesquisa. Este tipo de análise busca determinar se as respostas de participantes em diferentes subconjuntos de itens se relacionam mais entre si do que com outros subconjuntos, analisando a dimensionalidade entre os itens (KNEKTA et al., 2019).

Uma alternativa à teoria clássica é a Teoria de Resposta ao Item (TRI). A TRI fornece modelos matemáticos para representar a relação entre a probabilidade de um indivíduo dar uma certa resposta a um item (ARAUJO et al., 2009), fornecendo um cálculo de nota baseado nos itens individuais. Já que a teoria tradicionalmente utilizada, Teoria Clássica dos Testes (TCT), fornece a nota baseada na soma do número de questões da prova. A TRI permite identificar o grau da dificuldade de itens avaliativos e até definir uma escala de avaliação de desempenho (LALOR et al., 2016).

FEEDBACK E NOTAS

A avaliação pode ser usada para calcular notas e para fornecer *feedback* ao aluno (ALVES et al., 2019a). O resultado da avaliação informado ao aluno tipicamente se dá pelo uso de notas, indicando seu desempenho escolar. A atribuição de nota pode ser feita com um valor quantitativo ou qualitativo (ALVES et al., 2019a). Por exemplo, valores como "Bom", "Ótimo", ou "A", "B", "F" são qualitativos, faixas de um espectro definido que representam uma categoria de desempenho atingida pelo aluno. Já no quantitativo, a nota é composta por uma escala numérica, na qual alunos que desempenharam a tarefa com máximo aproveitamento recebem a nota máxima. No Brasil, não há uma padronização para a atribuição de notas. Cada Conselho Municipal de Educação possui autonomia para definir os instrumentos avaliativos, embora usualmente se dê pela alocação de notas de forma quantitativa por meio de números inteiros variáveis de 0 a 10 (CME, 2011).

A avaliação também pode servir para fornecer um *feedback* instrucional ao aluno, facilitando a ocorrência da aprendizagem (MORENO, 2004; KRETLOW; BARTHOLOMEW, 2010). O *feedback* pode ter grande contribuição para motivação dos alunos durante a realização das atividades instrucionais (NARCISS; HUTH, 2002), dado que estimula os alunos a refletirem sobre suas respostas, apresentando informações que possam servir de mudança na sua maneira de pensar e agir em relação ao conteúdo apresentado. Para isso, deve-se apresentar informações que sejam relevantes para o aluno, e entregá-las em momentos convenientes para sua aprendizagem (BILAL et al., 2012). Caso não seja elaborado dessa forma, o

feedback pode tornar-se incômodo e influenciar negativamente a aprendizagem e motivação do aluno.

Há duas principais abordagens para fornecer o *feedback*: por verificação, que avalia se as atividades efetuadas pelo aluno estão corretas ou não, e por elaboração, em que se fornece comentários e sugestões para o aluno, visando auxiliá-lo a identificar pontos a serem corrigidos por conta própria (BLACK; WILIAN, 1998; KRETLOW; BARTHOLOMEW, 2010). Ambas são importantes, já que a abordagem por verificação dá certezas ao aluno sobre sua aprendizagem, afetando positivamente sua atenção e motivação (BORDIA, 2004; ASHFORD, BLATT, & VANDEWALLE, 2003). A abordagem por elaboração pode analisar particularidades do aluno, indicando pontualmente erros e provendo informações sobre como corrigi-los (CHENG et al., 2005). Outra classificação avalia o *feedback* por sua corretude em três tipos, conforme apresentado na Tabela 7.

Tabela 7 - Classificação do feedback *por corretude (SHARIQ; PERERA, 2010; WANG et al., 2011)*

Tipo	**Descrição**
Positivo	Reconhece o sucesso obtido nas respostas corretas do aluno
Construtivo	Motiva a melhoria de respostas que não estão completas ou totalmente corretas
Negativo	Orienta o aluno a corrigir respostas incorretas

Com objetivo de deixar simultaneamente a avaliação mais leve e lúdica, é recomendado utilizar recursos multimídia, como imagens, vídeos e sistemas interativos. Uma técnica para atingir esse objetivo é utilizar-se de elementos de gamificação, que é a aplicação de elementos de *design* de jogos em atividades e contextos não relacionados a jogos (NAH et al., 2014). Pode-se citar como efeitos positivos da gamificação aplicada ao contexto educativo a sua capacidade de melhorar a retenção do conhecimento dos alunos e aprimorar suas habilidades sociais e práticas, como resolução de problemas, colaboração e comunicação (PUTZ et al., 2020).

Vários elementos são utilizados para aumentar o envolvimento dos alunos, como pontos, *badges*, *leaderboards* e níveis. O aluno pode receber pontos como recompensas, por exemplo, conforme vai realizando atividades que envolvam o processo de aprendizagem. As *badges* são emblemas que o aluno recebe ao realizar alguma atividade, alcançar um determinado somatório de pontos, etc. Colecionáveis e com identidade visual, são úteis para envolver os alunos em tarefas de aprendizagem subsequentes (NAH et al., 2014). *Leaderboards* são tabelas de classificação, usualmente ordenadas pela pontuação dos alunos, com objetivo de promover uma competitividade saudável e incentivar os alunos a promoverem seus nomes na tabela. Níveis são utilizados para dar aos alunos a sensação de progressão no processo de aprendizagem. Níveis iniciais tendem a exigir menos esforço e são mais rápidos de serem alcançados, enquanto níveis avançados exigem mais esforço e habilidades por parte dos alunos (NAH et al., 2014). Um exemplo de gamificação e seus elementos é a avaliação de projetos via CodeMaster (ALVES et al., 2019), em que o estudante é visualmente representado como um ninja que vai progredindo na obtenção de novas faixas conforme melhora no sistema de pontuação, como apresentado na Figura 6.

Figura 6 - Pontuação apresentada como feedback *lúdico na ferramenta CodeMaster (CNE, 2020)*

AUTOMAÇÃO DA AVALIAÇÃO

A automação dos métodos avaliativos permite um *feedback* rápido para os alunos, reduzindo a demanda de esforço e tempo dos avaliadores. Um grande volume de atividades torna impraticável a avaliação com um número reduzido de avaliadores, levando a um déficit de aprendizado para os alunos (GALAN et al., 2019). A automação das avaliações pode eliminar efeitos de cansaço nos educadores e injustificar alegações de favorecimento a determinados alunos na correção. Por seguir um modelo padronizado, formulado previamente com participação colaborativa dos educadores que aplicam a avaliação, é possível obter uma avaliação pouco enviesada e com maior confiabilidade.

O requisito básico para a avaliação automatizada de atividades é a mensurabilidade numérica dos alvos da avaliação, p.ex., questões de múltipla escolha ou artefatos computacionais, embora as abordagens semiautomáticas podem superar essa restrição (ALA-MUTKA, 2015).

Já existem abordagens para automatizar a avaliação de artefatos computacionais referentes à aprendizagem de algoritmos e programação (ALVES et al., 2019a). Para avaliação automatizada de artefatos de *software*, a análise pode ser dinâmica ou estática (ALVES et al., 2019a). A análise estática avalia características do *software* como estilo de código, erros de desenvolvimento, formatação, vulnerabilidades de segurança, etc. Já a análise dinâmica envolve a execução do *software*, podendo ser utilizado para validar diferentes entradas e saídas, geralmente por meio de intervalos para valores e heurísticas para questões qualitativas.

Porém, poucas abordagens de avaliação de desempenho fornecem sugestões de como melhorar o *software* ou utilizam recursos de gamificação. Há também ferramentas de automação de avaliação de *design* de interface de usuário (SOLECKI et al, 2020) e estética visual no contexto de desenvolvimento de aplicativos na Educação Básica (MARTINS, 2020).

3 ESTADO DA ARTE

Neste capítulo é apresentado o estado da arte atual de abordagens de avaliação de aprendizagem de *Machine Learning* no contexto do Ensino Médio. Para isto, é realizado um mapeamento sistemático da literatura seguindo o processo proposto por Petersen et al. (2008). Os resultados desta revisão do estado da arte foram também publicados em Salvador et al. (2021).

3.1 DEFINIÇÃO DO PROTOCOLO DE BUSCA

O mapeamento sistemático da literatura visa identificar, classificar e interpretar pesquisas disponíveis por meio de critérios de qualificação claros e reproduzíveis em relação ao tema deste trabalho (PETERSEN et al., 2008). A pergunta de pesquisa que este mapeamento procura responder é: **quais modelos existem para avaliação de aprendizagem de *Machine Learning* no Ensino Médio**? Esta questão é refinada nas seguintes perguntas de análise:

- **PA1.** Quais unidades instrucionais voltadas ao ensino de ML no Ensino Médio apresentam avaliações de aprendizagem?
- **PA2.** Quais são as características destas avaliações em termos de nível de aprendizagem, conteúdo e tipo?
- **PA3.** Que *feedback* instrucional é apresentado?

Bases de dados: São consideradas como fonte de pesquisa os artigos indexados pelas ferramentas digitais da área: Scopus, IEEE Xplore, ACM, SpringerLink, ScienceDirect, arXiv, SocArXiv e Google Scholar. Além disso, a pesquisa do Google foi utilizada para complementar a pesquisa, minimizando o risco de omitir algum modelo de avaliação de aprendizagem que não foi publicado como artigo científico (PIASECKI et al., 2018). Buscou-se também publicações no repositório do MIT Media Lab pela atuação nesta área.

Critérios de inclusão e exclusão: São considerados artigos na língua inglesa cujo foco seja apresentar um modelo de avaliação de aprendizagem de *Machine Learning* no contexto do Ensino Médio. A data de publicação considerada para os artigos é de 2011 a 2021, dado a tendência recente de ensino de conceitos de ML. Foram excluídos artigos com foco no ensino de ML no Ensino Fundamental e/ou Ensino Superior ou que abordam Inteligência Artificial sem abordar conceitos de ML. Publicações como blogs, vídeos ou outras ferramentas que não compõem uma unidade instrucional para estudantes de Ensino Médio e artigos que não apresentam informações substanciais foram descartados. Também foram excluídos artigos que somente apresentam uma avaliação do curso em si, mas não apresentam propostas para a avaliação da aprendizagem do aluno.

Critérios de qualidade: São considerados apenas artigos que apresentem informações substanciais para se extrair referentes às perguntas de análise. São excluídos artigos que apresentam, por exemplo, somente um resumo de uma proposta e para os quais não são encontradas mais informações detalhadas. Também são excluídas unidades instrucionais que possuem custo para acesso do material completo.

Termos de busca: Com base na pergunta de pesquisa foram definidos inicialmente os termos de busca e seus sinônimos indicados na Tabela 8.

Tabela 8 - Termos de busca iniciais

Termo	**Sinônimos**
machine learning	-
assessment	-
education	*teaching*
high school	*k-12*

Usando esses termos de busca, não foi possível obter resultados relevantes na execução da busca na literatura. A estratégia seguida foi reutilizar os termos e string de busca do mapeamento sistemático do ensino de *Machine Learning* realizado por Marques et al. (2020), como apresentado na Tabela 9. Buscou-se

ampliar as bases de dados pesquisadas, incluindo também possíveis artigos que foram publicados entre a data de publicação do mapeamento e este trabalho.

Tabela 9 - Termos de busca finais

Termo	**Sinônimos**
machine learning	*data science, artificial intelligence, deep learning*
education	*teach*, course, mooc, learn**
*school**	*k-12, kids, children, teen**

Uma *string* de busca foi definida a partir dos termos de busca para aplicar nas bases de dados.

```
(teach* OR education OR course OR MOOC OR learn*) AND (“machine learning” OR “data science” OR “artificial intelligence” OR “deep learning”) AND (“k-12” OR school* OR kids OR children OR teen*)
```

Com a *string* de busca definida, ela tem seu formato adaptado para realizar a busca em cada uma das bases de dados consideradas, conforme a Tabela 10.

Tabela 10 - Strings *de buscas utilizadas nas diferentes bases de dados*

Base de dados	***String* de busca**
Scopus	`(TITLE-ABS-KEY(teach* OR education OR course OR mooc OR learn*) AND TITLE-ABS-KEY("machine learning" OR "data science" OR "artificial intelligence" OR "deep learning") AND TITLE-ABS-KEY("k-12" OR school* OR kids OR children OR teen*)) AND PUBYEAR > 2010 (LIMIT-TO(SUBJAREA, "COMP"))`
IEEE Xplore	`("Abstract":teach* OR "Abstract":education OR "Abstract":course OR "Abstract":MOOC OR "Abstract":learn*) AND ("Abstract":“machine learning” OR "Abstract":“data science” OR "Abstract":“artificial intelligence” OR "Abstract":“deep learning”) AND ("Abstract":“k-12” OR "Abstract":school* OR "Abstract":kids OR "Abstract":children OR "Abstract":teen*)`
ACM	`[[Abstract: teach*] OR [Abstract: education] OR [Abstract: course] OR [Abstract: mooc] OR [Abstract: learn*]] AND [[Abstract: "machine learning"] OR [Abstract: "data science"] OR [Abstract: "artificial intelligence"] OR [Abstract: "deep learning"]] AND [[Abstract: "k-12"] OR [Abstract: school*] OR [Abstract: kids] OR [Abstract: children] OR [Abstract: teen*] OR [Abstract:))]] AND [Publication Date: (01/01/2011 TO *)]`

SpringerLink	(teach* OR education OR course OR MOOC OR learn*) (“machine learning” OR “data science” OR “artificial intelligence” OR “deep learning”) (“k-12” OR school* OR kids OR children OR teen*) Discipline: Computer Science 2011-2021
ScienceDirect	Year: 2011-2021 Title, abstract, keywords: (teach OR education OR course OR MOOC OR learn) (“machine learning” OR “data science” OR “artificial intelligence”) (“k-12” OR school OR teen)
arXiv	order: -announced_date_first; size: 200; date_range: from 2011-01-01 to 2021-12-31; classification: Computer Science (cs); include_cross_list: True; terms: AND abstract=teach* OR education OR course OR MOOC OR learn*; AND abstract=“machine learning” OR “data science” OR “artificial intelligence” OR “deep learning”; AND abstract=“k-12” OR school* OR kids OR children OR teen*
SocArXiv	(teach* OR education OR course OR MOOC OR learn*) (“machine learning” OR “data science” OR “artificial intelligence” OR “deep learning”) (“k-12” OR school* OR kids OR children OR teen*)
Google Scholar	(teach* OR education OR course OR MOOC OR learn*) (“machine learning” OR “data science” OR “artificial intelligence” OR “deep learning”) (“k-12” OR school* OR kids OR children OR teen*)
Google	“machine learning” teach (“K-12” OR school)

3.2 EXECUÇÃO DA BUSCA

A busca dos artigos foi realizada em fevereiro de 2021 pelo autor do presente trabalho e revisada pela orientadora. A busca inicial resultou em 92182 artigos, excluindo os resultados da pesquisa complementar no Google. A quantidade de artigos em cada etapa do processo de seleção é apresentada na Tabela 11.

Tabela 11 - Número de artigos identificados por base de dados

Base de dados	***Número de artigos identificados***	***Número de artigos analisados***	***Primeira etapa analisando título e resumo de curso de ML no Ensino Médio***	***Segunda etapa analisando o texto na íntegra de curso de ML no Ensino Médio***	***Terceira etapa analisando o texto na íntegra de avaliação como parte do curso de ML no Ensino Médio***
Scopus	3.692	500	6	3	0
IEEE Xplore	1.085	500	11	4	1
ACM	312	312	9	7	2
SpringerLink	47.238	500	4	1	0
ScienceDirect	57	57	1	0	-
arXiv	137	137	1	1	1
SocArXiv	7.209	500	4	2	1
Google Scholar	17.000	500	10	3	2
Google	70.700.000	500	42	14	11

MIT Media Lab	81	81	10	2	2
Total					**12 (sem duplicatas)**

Como o mapeamento inclui unidades instrucionais de ensino de *Machine Learning* cobrindo todos os anos escolares, somente os resultados que continham dados sobre Ensino Médio foram considerados. Destes, as informações foram coletadas apenas das atividades que envolvem e/ou possuem algum tipo de avaliação.

Na primeira etapa de análise, os títulos e resumos dos artigos foram analisados, resultando na remoção de artigos que não possuíam características levantadas no critério de inclusão. Na segunda etapa, os artigos foram analisados integralmente e, possuindo informações relevantes às respostas da pesquisa, considerados relevantes. Uma última etapa de seleção foi efetuada, selecionando apenas os cursos e atividades de ML para alunos do Ensino Médio que possuíam avaliações de ensino e informações relevantes sobre a avaliação.

Muitos artigos encontrados utilizam *Machine Learning* para a avaliação de ensino de alunos ou para outros propósitos educacionais, mas não se referem à avaliação de ensino de ML em si.

3.3 EXTRAÇÃO DAS INFORMAÇÕES

Dados foram sistematicamente extraídos dos artigos encontrados visando responder às questões de análise definidas no protocolo de busca. As informações relevantes às perguntas de análise foram extraídas conforme especificado na Tabela 12. Algumas informações foram inferidas pelo autor quando não presentes de maneira explícita nas referências.

Tabela 12 - Especificação das informações extraídas

Pergunta de análise	**Dados a extrair**
PA1. Quais avaliações de aprendizagem	Nome

de *Machine Learning* no Ensino Médio existem?	Descrição
	Fonte
PA2. Quais são as características destas avaliações?	Método de avaliação
	Níveis de aprendizado do Bloom
	Fase do *"Use-Modify-Create"*
	Conceitos de *Machine Learning*
PA3. Se e como o *feedback* instrucional é apresentado?	Tipo de *feedback* instrucional
	Tipo de automação

PA1. QUAIS UNIDADES INSTRUCIONAIS VOLTADAS AO ENSINO DE ML NO ENSINO MÉDIO APRESENTAM AVALIAÇÕES DE APRENDIZAGEM?

O primeiro passo para estudar os modelos de avaliações é encontrar as avaliações de aprendizagem existentes, além de mapear as referências, nome e fonte para consultas posteriores. Os dados sobre as avaliações são apresentados na Tabela 13.

Tabela 13 - Avaliações de aprendizagem de Machine Learning *no Ensino Médio*

Referência	**Nome**	**Descrição**	**Fonte**
(APPS FOR GOOD, 2019)	Apps for Good: ML course	Curso que fornece uma visão geral de diversos tópicos de ML e permite às equipes de alunos projetar e construir um protótipo que resolva um problema real usando algoritmos de ML	https://www.appsforgood.org/courses/machine-learning
(BRUMMELEN et al., 2020)	AI Literacy Workshop	Curso de IA a partir do qual os alunos desenvolvem agentes de conversação usando uma interface no MIT App Inventor.	https://arxiv.org/abs/2009.05653
(CODE.ORG, 2019)	AI for Oceans	Tutorial de introdução a *Machine Learning* envolvendo classificação de peixes do oceano	https://curriculum.code.org/hoc/plugged/9/
(CSER, 2020)	Teaching Artificial Intelligence in the Secondary Classroom	Curso gratuito de IA voltado para instrutores de IA usando ML em reconhecimento de imagens e visão computacional.	https://csermoocs.appspot.com/ai_secondary/
(ELEMENTS OF AI, 2019)	Elements of AI	Capítulo de ML de um curso online a aprender o que é Inteligência Artificial, o que pode (e não pode) ser feito com IA e como começar a criar métodos de IA	https://course.elementsofai.com/4
(EXPLORING	Artificial Intelligence	A unidade faz com que os alunos explorem as	http://www.exploringcs.or

COMPUTER SCIENCE, 2020)	Alternate Curriculum Unit	aplicações práticas diárias da IA que provavelmente terão um impacto em suas vidas por meio de diversas atividades	g/wp-content/uploads/2019/09/AI-Unit-9-16-19.pdf
(GRESSE VON WANGENHEIM et al., 2020)	*Machine Learning* para Todos	Curso que introduz conceitos básicos de ML, levando o aluno a criar um modelo de reconhecimento de imagens para separação de lixo reciclável	https://cursos.computacaonaescola.ufsc.br/cursos/curso-mlparatodos/
(LEE et al., 2021)	Developing Middle School Students' AI Literacy	Oficina de verão para preparar alunos do Ensino Médio a desenvolver conhecimentos e habilidades fundamentais sobre IA e tornarem cidadãos informados e consumidores críticos da tecnologia	https://aieducation.mit.edu/daily/
(MIT APP INVENTOR, 2019)	Introduction to Machine Learning: Image Classification	Curso que ensina noções básicas de aprendizado de máquina e a criação de aplicativos que implementam esses conceitos por meio de classificação de imagens pelos alunos	https://appinventor.mit.edu/explore/resources/ai/image-classification-look-extension
(READYAI, 2019)	ReadyAI AI + Me	É uma experiência online destinada a fornecer aos jovens alunos os conceitos básicos de IA	https://edu.readyai.org/courses/aime/
(RODRÍGUEZ-GARCÍA et al., 2021)	LearningML	Plataforma web gratuita para ensino de fundamentos de ML, permitindo treinar e testar modelos online. Formato de *workshop*	https://web.learningml.org/en/home-spanish-en-translation/
(TANG, 2019)	Personal Image Classifier	Curso que ensina noções básicas de aprendizado de máquina e faz com que os alunos criem seus próprios aplicativos que implementam esses conceitos por meio de classificação de imagens	https://appinventor.mit.edu/explore/resources/ai/personal-image-classifier

Observa-se que mesmo tendo uma quantidade considerável de cursos de *Machine Learning* para o Ensino Médio, poucos destes abordam de forma explícita a avaliação. Como resultado da pesquisa, 12 avaliações de aprendizagem de ML em atividades e cursos voltados ao Ensino Médio foram encontradas. No mapeamento sistemático levado como base, foram encontradas 12 unidades instrucionais de ensino de ML para o Ensino Médio. Dentre estas, 4 possuíam algum tipo de avaliação de aprendizagem. Fica evidente a falta de alguma forma de avaliação nestes cursos, etapa importante do processo de aprendizagem do aluno. Alguns desses cursos são especificamente voltados para ML e outros inserem ML dentro do contexto de Inteligência Artificial.

PA2. QUAIS SÃO AS CARACTERÍSTICAS DESTAS AVALIAÇÕES EM TERMOS DE NÍVEL DE APRENDIZAGEM, CONTEÚDO E TIPO?

É essencial entender quais são os tipos de avaliação utilizados, quais os níveis de aprendizagem da taxonomia de Bloom atingidos e quais fases do ciclo *"Use-Modify-Create"* foram contempladas. As características das avaliações são apresentadas na Tabela 14.

Tabela 14 - Características das avaliações de aprendizagem

Nome	**Método de avaliação**	**Níveis de aprendizagem do Bloom**	**Fases do *"Use-Modify-Create"***	**Conceitos de ML avaliados**
(APPS FOR GOOD, 2019)	Questões de avaliação	Conhecimento, Compreensão, Aplicação	*Use, Create*	Conceitos básicos de ML, impacto do ML
(BRUMMELEN et al., 2020)	Questões de avaliação, avaliação baseado em desempenho de ideas	Conhecimento, Compreensão e Aplicação	*Use-Modify-Create*	Conceitos básicos de ML, impacto de ML
(CODE.ORG, 2019)	Conclusão da tarefa	Conhecimento, Compreensão	--	Conceitos básicos de ML
(CSER, 2020)	Questões abertas/de redação	Conhecimento, Compreensão	*Use*	Conceitos básicos de ML
(ELEMENTS OF AI, 2019)	Quiz com 1-3 questões abertas/de redação ou de múltipla escolha	Conhecimento, Compreensão	--	Conceitos básicos de ML, impacto de ML, redes neurais
(EXPLORING COMPUTER SCIENCE, 2020)	Rubricas com critérios com 8 - 14 para a avaliação das apresentações dos alunos, questões de redação	Conhecimento, Compreensão	--	Conceitos básicos de ML, impacto de ML
(GRESSE VON WANGENHEIM et al., 2020)	Quiz, questões verdadeiro/falso, arrastar e soltar, rubrica de 11 critérios para a avaliação dos artefatos do desenvolvimento de um modelo de ML	Conhecimento, Compreensão, Aplicação	*Use*	Conceitos básicos de ML, redes neurais, processo de ML, impacto de ML
(LEE et al., 2021)	Quiz, questões verdadeiro/falso, questões abertas/de redação	Conhecimento, Compreensão	*Use, Modify*	Conceitos básicos de ML, redes neurais
(MIT APP INVENTOR, 2019)	Quiz com 3 questões de múltipla escolha	Conhecimento, Compreensão, Aplicação	*Use*	Redes neurais
(READYAI, 2019)	Quiz de pergunta única (múltipla escolha, arrastar e soltar, etc.)	Conhecimento, Compreensão	--	Conceitos básicos de ML, impacto de ML
(RODRÍGUEZ-GARCÍ	Quiz com 14	Conhecimento,	*Use, Modify, Create*	Conceitos básicos de

A et al., 2021)	questões, exercícios, questão aberta	Compreensão, Aplicação		ML, redes neurais
(TANG, 2019)	Quiz com 3 questões de múltipla escolha	Conhecimento, Compreensão, Aplicação	*Use*	Redes neurais

A maioria das avaliações são relativamente simples, referindo-se somente à aprendizagem nos níveis de Conhecimento e Compreensão, seguindo a Taxonomia de Bloom et al. (1965), com poucas que abordam também o nível de Aplicação. Como o foco da maioria dos cursos no nível introdutório, a aprendizagem de conceitos básicos de ML é a mais avaliada, menos frequentemente avalia-se a aprendizagem de conceitos de redes neurais. Várias avaliações abordam questões éticas e o impacto de ML na sociedade.

Em alguns casos, as avaliações consistem somente em questões de resposta única no final das unidades instrucionais ou apenas monitoram a conclusão da tarefa (CODE.ORG, 2019). Algumas unidades avaliam as respostas dos estudantes a exercícios (ELEMENTS OF AI, 2019). Várias unidades instrucionais usam *quizzes* com diferentes tipos de questões (verdadeiro/falso, múltipla escolha, arrastar e soltar, etc.) p.ex. (GRESSE VON WANGENHEIM et al., 2020; READYAI, 2019; RODRÍGUEZ-GARCÍA et al., 2021), enquanto MIT App Inventor (2019) e Tang (2019) utilizam testes com 3 questões de múltipla escolha. Ao levar em consideração que a maioria das unidades instrucionais são usualmente oferecidas como atividades extracurriculares, estes tipos de avaliações simples são adequados para prevenir a desmotivação dos estudantes. Entretanto, a falta de avaliações mais rigorosas pode impedir um melhor suporte para a aprendizagem dos alunos e o aprimoramento das unidades instrucionais.

Poucas unidades adotam uma avaliação baseada em desempenho definindo rubricas para a avaliação de apresentações dos projetos (EXPLORING COMPUTER SCIENCE, 2020) ou rubricas para avaliar artefatos de modelos de ML criados pelos alunos (GRESSE VON WANGENHEIM et al., 2020) Dado a natureza recente das unidades instrucionais de ML para o Ensino Médio, a maioria foca na avaliação de resultados na fase de *Use* do ciclo "*Use-Modify-Create*". Há exceções, como as unidades propostas por Rodríguez-García et al. (2021) e Apps for Good (2019),

levando os estudantes a criarem modelos personalizados de ML. Essa criação é feita adotando a estratégia de "ação computacional" (TISSENBAUM et al., 2019), que permite que os alunos aprendam enquanto criam artefatos significativos que têm um impacto em suas vidas e comunidades.

PA3. QUE *FEEDBACK* INSTRUCIONAL É APRESENTADO?

O *feedback* pode ser apresentado de diferentes formas em uma avaliação. Para permitir a avaliação em larga escala, de maneira consistente e confiável, é interessante que sua abordagem seja automatizada. Buscou-se encontrar então quais são os tipos de *feedbacks* instrucionais oferecidos aos estudantes e quais os tipos de automação da correção, caso exista. As informações são apresentadas na Tabela 15.

Tabela 15 - Conceitos de Machine Learning *avaliados*

Nome	**Tipo de *feedback***	**Tipo de automação**
(APPS FOR GOOD, 2019)	Avaliação manual pelo instrutor	Não possui
(BRUMMELEN et al., 2020)	Avaliação manual pelo instrutor	Não possui
(CODE.ORG, 2019)	Indicação de conclusão da tarefa sem análise de correção. *Feedback* para instrutores para monitoramento de uma turma	Grau de conclusão da tarefa
(CSER, 2020)	Comparação manual por parte do aluno de um modelo ideal fornecido pelo instrutor	Não possui
(ELEMENTS OF AI, 2019)	Correção de respostas de múltipla escolha, apresenta respostas de exemplo e revisão por pares para respostas de texto. Há certificado, e a quantidade de exercícios realizados é monitorada	Correção automatizada de respostas de múltipla escolha
(EXPLORING COMPUTER SCIENCE, 2020)	Avaliação manual pelo instrutor indicando o total de pontos	Não possui
(GRESSE VON WANGENHEIM et al., 2020)	Correção de respostas, acompanhamento de conclusão de tarefas	Correção automatizada de respostas dos *quizzes*, não possui automação para avaliação da rubrica
(LEE et al., 2021)	Avaliação manual pelo instrutor	Não possui
(MIT APP INVENTOR, 2019)	Avaliação manual pelo instrutor	Não possui
(READYAI, 2019)	Correção de respostas,	Correção automatizada de respostas

	acompanhamento de conclusão de tarefas	
(RODRÍGUEZ-GARCÍA et al., 2021)	Autocorreção testando o modelo	Cálculo de precisão e desempenho do modelo criado
(TANG, 2019)	Avaliação manual pelo instrutor	Não possui

O *feedback* instrucional, tipicamente, é limitado à indicação se as questões foram respondidas corretamente, sem maiores informações. Muitas avaliações não possuem algum tipo de automação para efetuar as avaliações, sendo executadas por instrutores das unidades instrucionais. Basicamente, somente *quizzes* são automatizados nas unidades online. Uma exceção é a avaliação apresentada por Rodríguez-García et al. (2021), que retorna a precisão e desempenho do modelo criado para a avaliação do aluno. A avaliação de Exploring Computer Science (2020) utiliza uma escala de pontuação para apresentar o *feedback* ao estudante com base na soma ponderada dos itens da rubrica, sendo julgado por juízes que representam especialistas do domínio. Ao final, algumas unidades oferecem um certificado aos estudantes que completam o curso para aumentar o engajamento (CODE.ORG, 2019).

Somente a avaliação de Artificial Intelligence Alternate Curriculum Unit (EXPLORING COMPUTER SCIENCE, 2020) utiliza-se de um sistema de pontuação para apresentar o *feedback* ao estudante. Há um sistema de pontos com diferentes pesos para cada item de uma atividade realizada com sucesso. Os pontos são somados conforme uma rubrica disponibilizada. Nenhuma adota uma abordagem de gamificação para aumentar a motivação do aluno também no momento da avaliação.

3.4 DISCUSSÃO

Observa-se como resultado desta revisão que atualmente a maioria das unidades instrucionais de ensino de *Machine Learning* para estudantes do Ensino Médio não apresentam propostas para a avaliação de aprendizagem dos alunos. Isto pode ser explicado pelo fato que o ensino de ML no Ensino Médio ainda está emergente com um aumento de unidades instrucionais principalmente a partir dos

últimos dois anos (MARQUES et al., 2020). Mesmo vários artigos apresentando avaliações da qualidade de curso por meio de pré/pós-testes, no quesito de avaliação de aprendizagem do aluno usam somente uma autoavaliação, sem medir a aprendizagem de uma forma mais confiável.

O foco da maioria dos cursos no nível iniciante explica a ênfase em níveis de aprendizagem mais baixos segundo a Taxonomia de Bloom, com poucos que levam e avaliam a competência ao nível de aplicação de ML. Assim, a maioria das avaliações foca em conceitos básicos a serem avaliados por meio de *quizzes* e testes. Foram identificados poucos exemplos de rubricas de avaliação de desempenho que analisam artefatos criados pelo aluno ao aplicar conceitos de ML. Outro fator que impede a avaliação de um escopo maior é a típica curta duração destes cursos, muitas vezes de forma extracurricular. O tipo de *feedback* apresentado para os alunos costuma ser simples, o qual indica se a tarefa foi concluída ou realizada corretamente. A maioria das avaliações propostas são realizadas manualmente pelos instrutores e/ou juízes. Identificou-se a automação de avaliações somente no caso de cursos online referente a *quizzes*.

Com base nestes resultados é evidente a necessidade de aprimoramento de modelos de avaliação do ensino de ML no Ensino Médio, tanto em termos do seu desenvolvimento e validação sistemático quanto uma cobertura maior dos objetivos de aprendizagem e automação para preparar uma futura adoção mais ampla em escolas brasileiras.

AMEAÇAS À VALIDADE DA REVISÃO DA LITERATURA

Como em qualquer mapeamento sistemático, existem algumas ameaças à validade dos resultados. As ameaças potenciais foram identificadas e estratégias de mitigação para minimizar os impactos foram aplicadas.

- **Viés de publicação**: Mapeamentos sistemáticos podem sofrer do viés comum de que os resultados positivos têm maior probabilidade de serem publicados do que os negativos. No entanto, foi considerado que os

resultados dos artigos, sejam positivos ou negativos, têm apenas uma pequena influência sobre esse mapeamento sistemático, uma vez que foi buscado caracterizar os modelos de avaliação de aprendizagem de *Machine Learning* no Ensino Médio.

- **Identificação de estudos**: Outro risco é a omissão de estudos relevantes. A fim de mitigar esse risco, a *string* de busca foi construída para ser o mais abrangente possível, considerando não apenas os principais conceitos, mas também sinônimos. Isso gera um possível problema em ocultar artigos em bases de dados que foram recortadas em 500 resultados pelo grande volume de artigos encontrados, embora a possibilidade de ocorrência de tal fato seja menor do que a de omitir resultados sem usar sinônimos.

- **Seleção e extração de dados de estudos**: Ameaças para estudar seleção e extração de dados foram mitigadas por meio do fornecimento de uma definição detalhada dos critérios de inclusão, exclusão e de qualidade. Foi definido e documentado um protocolo para a seleção do estudo e a execução do mapeamento foi validada pela orientadora do presente trabalho.

4 DESENVOLVIMENTO DO MODELO DE AVALIAÇÃO

Neste capítulo é apresentado o modelo de avaliação de aprendizagem de *Machine Learning* em cursos e atividades voltadas para o público do Ensino Médio, assim como a análise do contexto do público alvo e definição de expectativas em relação à avaliação. Também é desenvolvido o plano de avaliação, a rubrica e *feedback* instrucional. O processo de construção da avaliação é feito com base no *Evidence-Centered Design* (ECD), uma maneira sistemática de construir avaliações que visa garantir que a evidência coletada e interpretada é consistente com o conhecimento que a avaliação busca avaliar (MISLEVY et al., 2003).

O ECD é baseado no conceito de camadas lógicas, servindo como etapas na progressão da construção da avaliação. A primeira etapa, análise de domínio, consiste em levantar informações sobre o domínio a ser avaliado, como possíveis currículos, estratégias instrucionais a serem tomadas, como a informação é transmitida, etc. Com base nas informações, é possível seguir para a modelagem de domínio, definindo o construto a ser avaliado e selecionando aspectos relevantes do domínio (ZIEKY, 2014). No desenvolvimento do *framework* conceitual, são especificadas as atividades a serem realizadas e os modelos de medição (modelo de estudante, de evidência, de tarefa e de montagem). Na próxima fase de implementação, grande parte do material de teste é criado (ZIEKY, 2014). Após aplicar a avaliação, dados são coletados sobre os resultados.

4.1 ANÁLISE E MODELAGEM DO DOMÍNIO

Inicialmente, analisa-se o domínio do cenário em que pretende-se aplicar a avaliação, buscando garantir que será coerente com a realidade, não exigindo além do que o necessário dos avaliados.

Público-alvo. O modelo de avaliação de aprendizagem a ser desenvolvido é voltado para estudantes do Ensino Médio de escolas brasileiras. O Ensino Médio corresponde à última fase da Educação Básica, com duração de três anos e alunos na faixa etária regular de 15 a 18 anos. Considera-se que nessa faixa os alunos

possuem fluência na língua portuguesa do Brasil e conhecimentos básicos de línguas estrangeiras como o inglês e espanhol. É esperado que os estudantes dessa fase consolidem e aprofundem os conhecimentos adquiridos no Ensino Fundamental, obtenham preparo básico para o mercado de trabalho, além de compreender fundamentos científico-tecnológicos dos processos produtivos, relacionando a teoria com a prática no ensino de cada disciplina (MINISTÉRIO DA EDUCAÇÃO, 2017). As atividades e trabalhos que os estudantes se envolvem exigem domínio de conhecimentos específicos com maior complexidade, além do currículo apresentar maior número de disciplinas (MINISTÉRIO DA EDUCAÇÃO, 2018).

O conhecimento adquirido no Ensino Fundamental trata a tecnologia unicamente como recurso. Dada a intrínseca relação entre culturas juvenis e cultura digital e o recente avanço nas tecnologias de informação e comunicação, é natural que os jovens estejam inseridos no contexto tecnológico, não somente como consumidores, mas cada vez mais aparecendo como protagonistas (MINISTÉRIO DA EDUCAÇÃO, 2017). O uso de tecnologias como dispositivos móveis para executar atividades escolares é bem disseminado nesta fase, conforme Figura 7. Usualmente esses estudantes possuem acesso à internet em seu domicílio, como apresentado na Figura 8. Os estudantes do Ensino Médio, de acordo com currículo proposto pela SBC (SBC, 2018a), teoricamente devem ter contato com os fundamentos da Inteligência Artificial e Robótica, não envolvendo *Machine Learning*. Assim, na prática, abordagens de ensino de ML para estudantes do Ensino Médio são praticamente inexistentes.

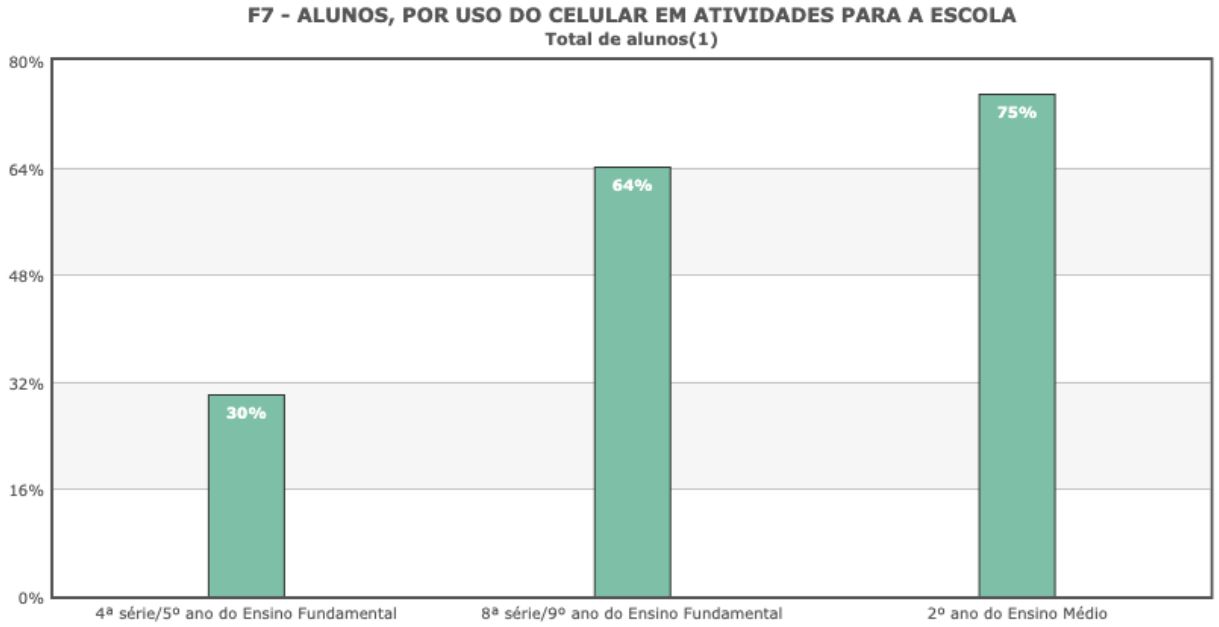


Figura 7 - Porcentagem de alunos por uso do celular em atividades para a escola (CETIC.BR, 2017)

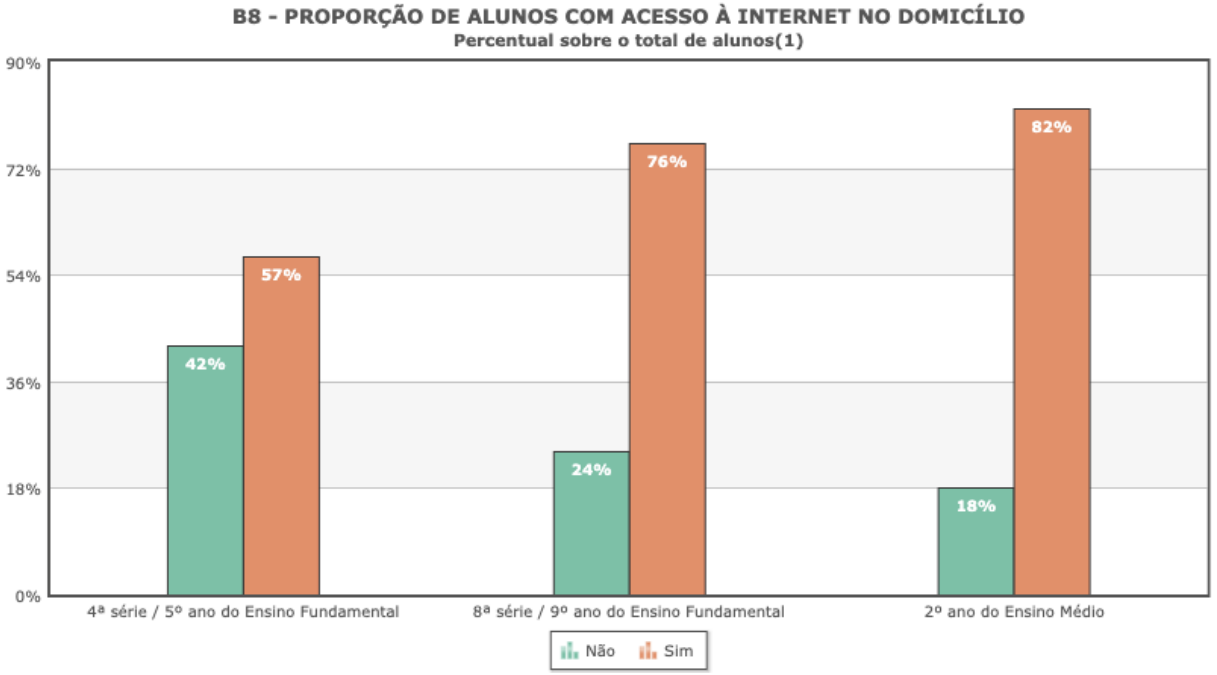


Figura 8 - Proporção de alunos com acesso à internet no domicílio (CETIC.BR, 2014)

Contexto escolar. A disponibilidade de recursos tecnológicos (laboratório de informática, Internet e Internet banda larga) nas escolas de Ensino Médio é maior do que a observada para o Ensino Fundamental. Esses recursos são encontrados em mais de 60% das escolas (INEP, 2018).

As salas de informática são geralmente coordenadas por um professor auxiliar de tecnologia educacional. Esse profissional pode ser graduado ou graduando em algum curso de licenciatura relacionado à tecnologia ou em pedagogia com alguma especialização na área tecnológica. As demais disciplinas são ministradas por professores da respectiva área. Atualmente, porém, não existem profissionais suficientes com qualificação para atuar nas aulas de computação. Por exemplo, a maioria dos professores de escolas urbanas nunca receberam capacitação para uso de computador e Internet em atividades de ensino-aprendizagem, conforme apresentado pela Figura 9.

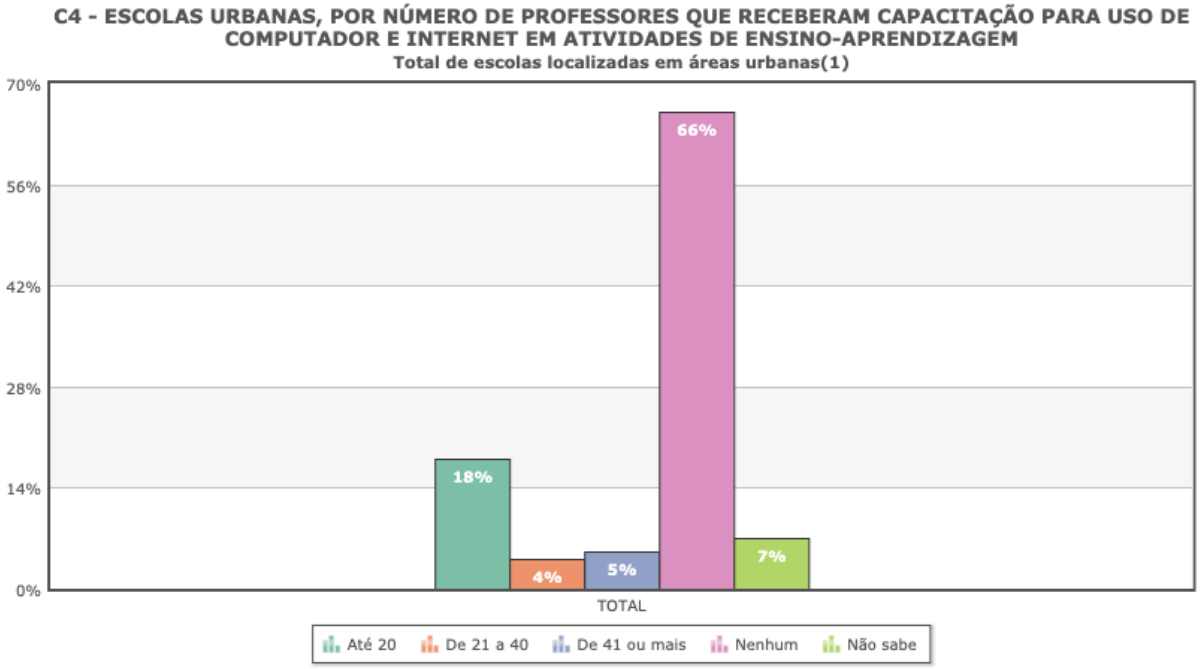


Figura 9 - Escolas urbanas, por número de professores que receberam capacitação para usar informática em atividades de ensino-aprendizagem (CETIC.BR, 2017)

Ensino de *Machine Learning* com Jupyter. O modelo de avaliação de aprendizagem desenvolvido no presente trabalho é aplicado em um curso de *Machine Learning* voltado a classificação de imagens de árvores nativas de Santa Catarina (CARDOZO, 2021). O curso é interativo, online e pode ser realizado individualmente ou dentro de um contexto escolar/extracurricular com turma e acompanhamento de instrutores. O curso aborda as fases *Use* e *Modify* do ciclo *"Use-Modify-Create"*, com os objetivos gerais e conteúdo conforme a Tabela 16.

Tabela 16 - Visão geral do curso

Estágio	***Use***	***Modify***
Objetivo geral	Desenvolver um app inteligente para classificação de 6 árvores nativas da Grande Florianópolis/SC seguindo um tutorial predefinido	O aluno explora como modificar o app desenvolvido, como ao acrescentar mais tipos de árvores, incluindo descrição das árvores ou mudando o design da interface
Conteúdo	Processo de desenvolvimento de ML: preparação dos dados, criação do modelo ML, *deploy* em app usando Jupyter/Colab	Diferentes processos, dependendo do tipo da aplicação

O curso aborda um tema interdisciplinar com a disciplina de Biologia voltado a classificação de imagens de árvores nativas de Santa Catarina. As árvores nativas a serem classificadas foram escolhidas com base no inventário florístico do campus Trindade da Universidade Federal de Santa Catarina, em Florianópolis. Seis espécies foram definidas em conjunto com pesquisadores da área de Biologia da UFSC, conforme a Tabela 17.

Tabela 17 - Espécies de árvores nativas utilizadas no curso

Nome popular	**Espécie**	**Família**	**Origem**
Aroeira-vermelha	*Schinus terebinthifolia*	Anacardiaceae	Nativa
Jerivá	*Syagrus romanzoffiana*	Arecaceae	Nativa
Mulungu, Eritrina	*Erythrina speciosa*	Fabaceae	Endêmica
Capororoca	*Myrsine guianensis*	Primulaceae	Nativa
Embaúba	*Cecropia glaziovii*	Urticaceae	Endêmica
Pitangueira	*Eugenia uniflora*	Myrtaceae	Nativa

O curso aborda uma parte introdutória apresentando conceitos básicos e exemplos de aplicação de ML. Seguindo o processo de ML os alunos são levados a desenvolver um modelo de ML, incluindo a preparação do conjunto de dados, etapas como o *upload* do conjunto de dados e separação de conjunto de dados de treinamento e validação. Os alunos também definem os parâmetros, treinam e avaliam o modelo usando *Transfer Learning* e *Fine-Tuning* utilizando o *framework*

fastai (HOWARD; GUGGER, 2020) em Jupyter Notebooks no ambiente do Google Colab. Ao final do curso, também são discutidas questões de ética e de impactos de ML na sociedade.

4.2 DESENVOLVIMENTO DO *FRAMEWORK* CONCEITUAL

Com base nas informações sobre o domínio da avaliação a ser criada, define-se um plano especificando as atividades a serem realizadas por meio de modelos de medição incluindo os modelos de estudante, de evidência e de tarefa).

4.2.1 MODELO DE ESTUDANTE

O modelo de estudante define uma ou mais variáveis relacionadas ao conhecimento sendo avaliado (MISLEVY et al., 2003), servindo de definição para o quê se deseja avaliar. No presente trabalho, as variáveis são os objetivos de aprendizagem esperados que os alunos atinjam durante o curso de *Machine Learning*, apresentados na Tabela 18. Os objetivos foram definidos em conformidade com as diretrizes de (AI4K12, 2020) referentes à grande ideia 3 - Aprendizagem, alfabetização em Inteligência Artificial de Long e Magerko (2020), tópicos gerais de computação propostos por (CSTA, 2017), e em conformidade como processo de ML por (AMERSHI et al., 2019). De forma interdisciplinar, inclui também um objetivo de aprendizagem relacionado a botânica como parte da disciplina de Biologia (MINISTÉRIO DA EDUCAÇÃO, 2017).

Tabela 18 - Objetivos de aprendizagem de Machine Learning *no Ensino Médio*

Categoria	**Objetivo de Aprendizagem**	**Nível de aprendizagem com base na Taxonomia Bloom**	**Referência(s)**
OA1. Conceitos básicos de ML	Identificar exemplos de *Machine Learning* e diferenciá-lo da aprendizagem humana	Lembrar	3-A-i K-2, 3-A-i 3-5, 3-A-i 6-8 (AI4K12, 2020); 1, 2, 3, 5 (LONG; MAGERKO, 2020)

OA2. Redes neurais	Compreender a estrutura de uma rede neural e descrever como suas partes formam um conjunto de funções que computam uma saída capaz de identificar padrões em dados	Compreender	3-A-ii 3-5, 3-B-i 6-8, 3-B-ii 3-5 (AI4K12, 2020)
OA3. Gerenciamento de dados	Preparar um conjunto de dados usado para treinar um modelo de ML considerando o tamanho do conjunto de dados, a forma com que os dados foram coletados e rotulados, além de sua qualidade (equilíbrio, balanceamento, viés)	Aplicar	3-C-ii 9-12 (AI4K12, 2020); 11, 12 (LONG; MAGERKO, 2020); 1A-DA-05 (CSTA, 2017)
OA4. Treinamento de modelo de ML	Treinar um modelo de ML para classificação/predição usando um algoritmo de aprendizagem supervisionada com dados reais e ajustando os parâmetros de treinamento	Aplicar	3-A-ii 9-12, 3-A-iii 9-12 (AI4K12, 2020)
OA5. Avaliação e interpretação do desempenho de um modelo de ML	Analisar e interpretar o desempenho de um modelo de ML para classificação/predição	Aplicar	(AMERSHI et al., 2019)
OA6. Implantação de um modelo de ML*	Exportar um modelo e integrar o modelo dentro de um sistema de software	Aplicar	(AMERSHI et al., 2019)
OA7. Processo de ML	Compreender e aplicar as etapas envolvidas no *Machine Learning* e suas práticas e desafios	Aplicar	3-A-iv 9-12 (AI4K12, 2020); 9 (LONG; MAGERKO, 2020)
OA8. Ética de ML	Identificar e descrever diferentes questões éticas acerca de ML (privacidade, viés introduzido por características dos dados de treinamento, tomadas de decisões éticas, etc.)	Compreender	3-C-iii 6-8 (AI4K12, 2020); 3A-AP-24 (CSTA, 2017); 13, 16 (LONG; MAGERKO, 2020)
OA9. Impactos do IA/ML	Identificar prós e contras de IA e ML para atividades cotidianas e opções de carreira atuais e futuras	Compreender	2-IC-21 (CSTA, 2017); 6 (LONG; MAGERKO, 2020)
OA10. Criar/modificar programas	Criar programas de computador usando sequências, eventos e outros comandos ou modificar programas existentes	Aplicar	1B-AP-10, 1B-AP-12 (CSTA, 2017)
OA11. Testar e aperfeiçoar programas	Testar e aperfeiçoar artefatos computacionais	Aplicar	1B-AP-15, 2-DA-09, 3A-IC-25 (CSTA, 2017)
OA12. Botânica	Reconhecer espécies de árvores nativas de Santa Catarina	Lembrar	(MINISTÉRIO DA EDUCAÇÃO, 2017)

* Como atualmente ainda não existe uma integração de modelo de ML criado com fastai e o App Inventor, este objetivo de aprendizagem embora previsto de forma geral, não é abordado neste TCC.

4.2.2 MODELO DE EVIDÊNCIA

O modelo de evidência é baseado em comportamentos ou produtos observáveis resultantes de respostas a uma tarefa específica (ZIEKY, 2014), apresentando como os objetivos mensurados devem ser avaliados de acordo com determinado desempenho atingido. O modelo de avaliação é composto incluindo diferentes tipos de avaliações como *quizzes* referentes aos conceitos básicos e questões de impacto/ética de ML (Tabela 19), além de uma avaliação de desempenho por meio de uma rubrica de avaliação (Tabela 20) com base nos artefatos criados pelos alunos durante o processo de ML.

Para o plano de avaliação, propostas de avaliação com seus respectivos tipos são fornecidas, assim como propostas de automação para o modelo de avaliação. O plano de avaliação referente aos quizzes não será automatizado neste trabalho, dado que o conteúdo do curso ainda está em desenvolvimento (CARDOZO, 2021). A Tabela 19 apresenta o plano de avaliação referente aos *quizzes* em relação aos objetivos de aprendizagem.

Tabela 19 - Plano de avaliação referente aos quizzes

ID	Critério de avaliação	Fonte	Proposta de avaliação
Conceitos básicos de ML (OA1)			
C1	Descrever e fornecer exemplos de como pessoas e computadores aprendem	3-A-i K-2 (AI4K12, 2020)	Redação
C2	Diferenciar entre como pessoas e computadores aprendem	3-A-i 3-5 (AI4K12, 2020)	Verdadeiro/falso
C3	Contrastar as características únicas do aprendizado humano e das maneiras que máquinas operam	3-A-i 6-8 (AI4K12, 2020)	Arrastar e soltar
C4	Distinguir entre artefatos tecnológicos que usam ou não IA	1 (Long & Magerko, 2020)	Arrastar e soltar
C5	Analisar e discutir criticamente características que fazem uma entidade “inteligente”, incluindo discutir diferenças entre humanos, animais e inteligência de máquina	2 (Long & Magerko, 2020)	Múltipla escolha
C6	Reconhecer que existem muitos jeitos diferentes de pensar e desenvolver máquinas “inteligentes”. Identificar uma variedade de tecnologias que usam IA, incluindo sistemas	3 (Long & Magerko, 2020)	Verdadeiro/falso

	cognitivos, robótica e ML		
C7	Identificar tipos de problemas que IA se sobressai e problemas que são mais difíceis para a mesma. Usar essa informação para determinar quando é apropriado usar IA e quando usar habilidades humanas	5 (Long & Magerko, 2020)	Verdadeiro/falso
Redes neurais (OA2)			
C8	Modelar como aprendizagem supervisionada identifica padrões em dados rotulados	3-A-ii 3-5 (A4IK12, 2020)	Verdadeiro/falso
C9	Ilustrar a estrutura de uma rede neural e descrever como suas partes formam um conjunto de funções que computam uma saída	3-B-i 6-8 (AI4K12, 2020)	Arrastar e soltar
C10	Demonstrar como pesos são designados em uma rede neural para produzir o comportamento desejado de entrada e saída	3-B-ii 3-5 (AI4K12, 2020)	Múltipla escolha
Processo de ML (OA7)			
C11	Ilustrar o que acontece em cada etapa necessária ao usar ML para construir um classificador ou preditor	3-A-iv 9-12 (AI4K12, 2020)	Arrastar e soltar
C12	Entender as etapas envolvidas no *Machine Learning* e suas práticas e desafios	9 (LONG; MAGERKO, 2020)	Múltipla escolha
Ética de ML (OA8)			
C13	Explicar como a escolha dos dados de treino molda o comportamento do classificador, e como esse viés pode ser introduzido se o conjunto de treino não for balanceado apropriadamente	3-C-iii 6-8 (AI4K12)	Redação
C14	Avaliar as formas em que a computação afeta as práticas pessoais, éticas, sociais, econômicas e culturais	3A-AP-24 (CSTA, 2017)	Múltipla escolha
C15	Entender que os dados não devem ser considerados pelo seu valor bruto e requerem interpretação; Descrever como os conjuntos de treinamento podem afetar os resultados de um algoritmo	13 (Long & Magerko, 2020)	Múltipla escolha
C16	Identificar e descrever diferentes perspectivas no problemas éticos chave acerca de IA (privacidade, tomadas de decisões éticas, etc.)	16 (Long & Magerko, 2020)	Redação
Impactos do IA/ML (OA9)			
C17	Comparar os prós e contras associados às tecnologias de computação que afetam as atividades cotidianas das pessoas e as opções de carreira	2-IC-21 (CSTA, 2017)	Múltipla escolha
C18	Imaginar possíveis aplicações futuras de IA e considerar os efeitos de tal aplicação no mundo	6 (Long & Magerko, 2020)	Redação

A pontuação dos *quizzes* segue os níveis de desempenho nas respostas: 0 pt. para incorretas, 1 pt. para parcialmente corretas e 2 pt. para corretas. Além dos *quizzes*, será realizada a avaliação de desempenho com base na rubrica de desempenho conforme a Tabela 20. Visa-se realizar a automação da avaliação de desempenho por meio da coleta de dados diretamente no Jupyter Notebook, como proposto na coluna "Proposta de automação" da Tabela 20.

Tabela 20 - Rubrica de avaliação e proposta de automação da avaliação

ID	**Critério**	**Níveis de desempenho**			**Técnica de automação**
		Baixo - 0 pt.	***Aceitável - 1 pt.***	***Bom - 2 pt.***	
Preparação de dados (OA3)					
C1	Quantidade de imagens	Menos de 5 imagens por categoria	6 de 10 imagens por categoria	Mais de 10 imagens por categoria	Listar os diretórios e somar a quantidade de arquivos de imagem a partir dos dados enviados ao *Jupyter Notebook*
C2	Distribuição do conjunto de dados	Quantidade de imagens por categoria varia muito	Quantidade de imagens por categoria varia pouco	Todas as categorias possuem a mesma quantidade de imagens	Listar os diretórios, somar a quantidade de arquivos de imagem e comparar variação por categoria a partir dos dados enviados ao *Jupyter Notebook*
Preparação de dados/Botânica (OA3/OA12)					
C3	Rotulagem das imagens	Menos de 20% das imagens rotuladas corretamente	De 20% a 99% das imagens rotuladas corretamente	Todas as imagens rotuladas corretamente	Aplicar um modelo de alta precisão para a tarefa de classificação das 6 categorias de árvores a partir dos dados enviados ao *Jupyter Notebook*
Treinamento de modelo de ML/*Transfer Learning* e *Fine-Tuning* (OA4)					
C4	Treinamento - *Transfer Learning*	O modelo não foi treinado (*transfer learned*)	O modelo foi treinado com os parâmetros padrão	O modelo foi treinado com parâmetros ajustados (arquitetura, época e taxa de aprendizagem)	Verificar se o comando de treinamento foi executado, retirar qual foi a arquitetura do comando de definição do learner, retirar qual o número de épocas e taxa de aprendizagem do comando de treinamento
C5	Treinamento - *Fine-Tuning*	O modelo não foi *fine-tuned*	Foi feito *unfreeze* das camadas e melhor taxa de aprendizagem não encontrada ou modelo não treinado	Foi feito *unfreeze* das camadas, a melhor taxa de aprendizagem foi encontrada e o modelo foi *fine-tuned*	Verificar se o comando de *unfreeze*, verificar se foi procurada a melhor taxa de aprendizagem (podendo ser qualquer um dos *valley*, *steep* etc.), se foi realizado o treinamento de *Fine-Tuning* com uma destas melhores taxa de aprendizagem encontradas
Avaliação e interpretação do desempenho de um modelo de ML (*Transfer Learning* e *Fine-Tuning*) (OA5)					
C6	Interpretação de acurácia	Categorias com baixa acurácia não	Categorias com baixa acurácia identificadas e	Categorias com baixa acurácia identificadas	Verificar se o comando de *display* de categorias com baixa acurácia foi executado e se respostas no *widget* de

		identificadas	interpretação incorreta em relação ao modelo	corretamente e interpretação correta em relação ao modelo	interpretação usando *ipywidgets* estão corretas
C7	Interpretação da matriz de confusão	Classificações incorretas não identificadas e interpretação incorreta em relação ao modelo	Classificações incorretas identificadas e interpretação incorreta em relação ao modelo	Classificações incorretas identificadas e interpretação correta em relação ao modelo	Verificar se o comando de display da matriz de confusão foi executado e se respostas no *widget* de interpretação usando *ipywidgets* estão corretas
C8	Ajustes/melhorias feitas	Sem novas iterações de desenvolvimento	Uma nova iteração com alterações no conjunto de dados e/ou parâmetros de treinamento	Diversas novas iterações com alterações no conjunto de dados e/ou parâmetros de treinamento	Verificar quantidade de iterações em que houve alterações no conjunto de dados e/ou parâmetros de treinamento
Avaliação e interpretação do desempenho de um modelo de ML/Testar e aperfeiçoar programas (OA5/OA11)					
C9	Testes com novos objetos	Nenhum novo objeto testado	1-2 novos objetos testados	Mais de dois novos objetos testados	Verificar quantidade de vezes que o comando de treinamento do modelo foi executado com novos objetos
C10	Interpretação dos testes	Interpretação errada	---	Interpretação correta	Verificar se respostas no *widget* de interpretação usando *ipywidgets* estão corretas
Criar/modificar programas (OA10)					
C11	Criação de programas	Programa não criado	Programa criado com sequências	Programa criado com sequências, eventos e outros comandos	*A definir, assim que o desenvolvimento da implantação do curso for finalizado*
C12	Modificação de programas	Partes de um programa existente não modificadas e incorporadas no programa criado	Partes de um programa existente incorporadas no programa, sem adicionar recursos mais avançados	Partes de um programa existente modificadas/incorporadas no programa criado para adicionar recursos mais avançados	*A definir, assim que o desenvolvimento da implantação do curso for finalizado*

Para verificar as respostas das questões de interpretação e coletar dados referentes ao treinamento dentro do Jupyter Notebook serão usado *widgets* desenvolvidos com a ferramenta ipywidgets (IPYWIDGETS, 2017)

Modelo de medição. Para o modelo de medição, é feito um cálculo para chegar a uma nota final para as atividades realizadas pelo estudante no curso. O cálculo é baseado na soma das pontuações dos *quizzes* e pontos da rubrica de desempenho.

A nota final do desempenho é dada em uma escala numérica ascendente, de 0 a 10, onde 10 representa o desempenho máximo esperado do estudante. O cálculo é feito da seguinte forma:

```
nota final = (soma da pontuação dos 10 critérios) / 2
```

4.2.3 MODELO DE TAREFA

O modelo de tarefa descreve o material que o avaliado produzirá, determinando onde o modelo de avaliação é aplicado. O modelo de tarefa da avaliação pelo desempenho refere-se ao desenvolvimento de um modelo de ML para a classificação de 6 espécies de árvores nativas de SC. Nesta tarefa o estudante deve desenvolver este modelo seguindo o tutorial apresentado no curso de ML (CARDOZO, 2021) em um Jupyter Notebook executado no Google Colab. É disponibilizado também para o estudante um conjunto de dados de 215 imagens de árvores não rotuladas.

A definição da tarefa é apresentada de forma detalhada na Tabela 21.

Tabela 21 - Definição do modelo de tarefa

Objetivo do modelo de DL	
Tarefa	Classificar a espécie de árvore de uma imagem de árvore (tipicamente a vista da árvore toda ou partes dentro do habitat natural (rua, praça, parque etc.) capturada de um aplicativo Android em relação a 6 categorias de árvores nativas/endêmicas de SC/Brasil.
Tipo da tarefa	Single-label classificação de imagens

Categorias	6 categorias de espécies de árvores nativas/endêmicas de SC/Brasil	Aroeira-vermelha, Capororoca, Embaúba, Jerivá, Mulungu, Pitangueira
Experiência	Conjunto de imagens de árvores (tipicamente a vista da árvore toda ou partes dentro do habitat natural (rua, praça, parque etc.)	
Fonte de dados	Conjunto de dados de árvores disponibilizado pela CnE	
Quantidade de dados	No total 215 imagens	
Padronização das imagens	Formato: .jpeg	Tamanho: 224x224 pixels
Rotulação de dados	A ser feito pelos estudantes do ensino médio	
Desempenho	O modelo de ML será otimizado para precisão para reduzir o risco de indicar a espécie errada ao usuário, levando ele a uma compreensão errada	
	Medido pela precisão e acurácia de no mínimo 0.75 total e por categoria de espécie de uma imagem em relação ao valor verdadeiro definido por biólogos	
Medidas	Acurácia (total/por categoria)	No mínimo 0.75
	Precisão	No mínimo 0.75

4.2.4 MODELO DE DOCUMENTAÇÃO

Com o objetivo de automatizar a documentação das principais características do modelo de ML criado pelo estudante é especificado um Cartão de Modelo com base em propostas similares (MITCHELL et al., 2019). O artefato é automaticamente gerado a partir das informações do Jupyter Notebook.

A proposta do Cartão de Modelo é servir como um procedimento padronizado de documentação de modelo. Ele centraliza informações sobre um modelo de ML treinado, p.ex., como foi construído, quais suposições foram feitas durante seu desenvolvimento, quais algoritmos foram utilizados, quais as características do conjunto de dados utilizados, etc (MITCHELL et al., 2019). O Cartão de Modelo a ser montado terá o formato conforme definido na Tabela 22.

Tabela 22 - Formato do Cartão de Modelo

Summary sheet - Modelo de ML	
Nome do modelo	
Data	
Versão	
Objetivo do modelo de ML	
Tarefa	
Contexto de uso	
Público alvo	
Riscos	
Tipo da tarefa	
Categorias	
Conjunto de dados	
Descrição dos dados	
Origem dos dados (coleta própria/uso de conjunto de dados pré-existente, p.ex. do kaggle)	
Quantidade total de dados	
Distribuição dos dados por categoria	
Labeling	
Tipos de aumento de dados aplicados (tipo rotate, crop, etc.)	
Tamanho de imagens	
Dataset splitting	
Treinamento - Transfer learning	
Tipo de modelo	
Tamanho do batch	
Quantidade de épocas	
Taxa de aprendizagem	
Avaliação - Transfer learning	
Acurácia por categoria	

Matriz de confusão	
Top x losses	
Treinamento - Fine Tuning	
Taxa de aprendizagem	
Quantidade de épocas	
Avaliação - Fine Tuning	
Acurácia total	
Acurácia por categoria	
Matriz de confusão	
Top x losses	
Predição	
Quantidade de testes realizados (via upload de imagens novas)	
Limitações e considerações éticas	
Limitações	
Considerações éticas referente aos dados	
Referências	
Autores e afiliação	

5 DESENVOLVIMENTO DA AUTOMAÇÃO

A partir da análise e modelagem do domínio e do *framework* conceitual desenvolvido, foram realizadas várias iterações para automatizar a rubrica de avaliação. Para esse fim, foi utilizado um processo iterativo/incremental de desenvolvimento de software (LARMAN; BASILI, 2003).

5.1 ANÁLISE DE REQUISITOS

A Tabela 23 apresenta os requisitos funcionais da automação e a Tabela 24 apresenta os requisitos não funcionais, identificados conforme a necessidade de aplicar a rubrica em Jupyter Notebooks no ambiente Google Colab.

Tabela 23 - Requisitos funcionais

Requisito	**Descrição**	**Artefato Entrada**	**Artefato Saída**
Automatizar a aplicação da rubrica de avaliação	A ferramenta deve automatizar a aplicação da rubrica de avaliação de aprendizagem de ML	Objeto com as informações do treinamento no Jupyter Notebook	Nota de 0 a 10 representando a aprendizagem de ML e pontuação por critério avaliado
Gerar um JSON com informações sobre o treinamento no *Notebook*	A ferramenta deve gerar um arquivo no formato JSON contendo as informações do Jupyter Notebook	Objeto com as informações do treinamento no Jupyter Notebook	Arquivo JSON com as informações do Jupyter Notebook
Gerar um Modelo de Documentação	A ferramenta deve gerar um Modelo de Documentação no formato PDF contendo as informações do Jupyter Notebook	Objeto com as informações do treinamento no Jupyter Notebook	Arquivo PDF no formato Modelo de Documentação com as informações do Jupyter Notebook
Apresentar um retorno visual dos resultados do modelo	A ferramenta deve retornar uma imagem de um “ninja-robô” retornando visualmente os resultados da avaliação de forma lúdica	Nota da avaliação	Imagem do “ninja-robô” com uma faixa conforme a nota da avaliação

Tabela 24 - Requisitos não funcionais

Requisito	**Descrição**
Linguagem de programação Python	Como a aplicação será executada em um Jupyter Notebook no Google Colab utilizando Python (recomenda-se a versão 3), a automação da rubrica deverá ser criada utilizando a mesma linguagem para compatibilidade de bibliotecas.
Repositório PyPI	A aplicação deve ser disponibilizada no repositório oficial de software de terceiros para Python, o PyPI (PYPI, 2021). É fonte padrão para pacotes e dependências do pip, disponível no Google Colab
Servidor HTTP	A aplicação deve ser disponibilizada como uma servidor HTTP, expondo uma rota para realizar a avaliação com base no JSON gerado com as informações do treinamento
Repositório Git	O código-fonte da aplicação deve estar disponibilizada em um repositório Git institucional

5.2 ARQUITETURA

A automação da avaliação implementada neste trabalho na forma de um pacote Python chamado *jupiclass* (***Jup****yter Notebook* ***I****mage* ***Cl****assification* ***ass****essment*). As informações sobre o conjunto de dados, modelo, treinamento do *Transfer Learning* e *Fine-Tuning* e respostas das questões de interpretação devem ser fornecidas em um objeto Python da classe *ImageClassification*. Esse objeto deve estar disponível e com seus atributos preenchidos no código Python do Jupyter Notebook em que será aplicada a automação da rubrica. O *jupiclass* possui quatro módulos: *grader*, *json*, *ninja_robot* e *pdf* e duas interfaces principais de uso: via pacote disponibilizado no PyPI ou servidor HTTP em Flask (2021), disponível para execução como container Docker (2021). O primeiro expõe a avaliação por uma função e o segundo por uma rota. Os módulos do *jupiclass* são apresentados a seguir.

- *grader*: responsável por automatizar a avaliação das competências, retornando o resultado da avaliação em um dicionário para o Jupyter Notebook;

- *json*: retorna um JSON com as informações coletadas do objeto *ImageClassification*. Esse JSON pode ser utilizado como corpo da requisição para a API;
- *ninja_robot*: salva no diretório uma imagem com um “ninja-robô” com uma faixa representando a nota atingida na avaliação;
- *pdf*: monta um Modelo de Documentação (conforme formato da seção 4) com as informações do modelo em formato PDF.

A arquitetura completa da automação é apresentada na Figura 10.

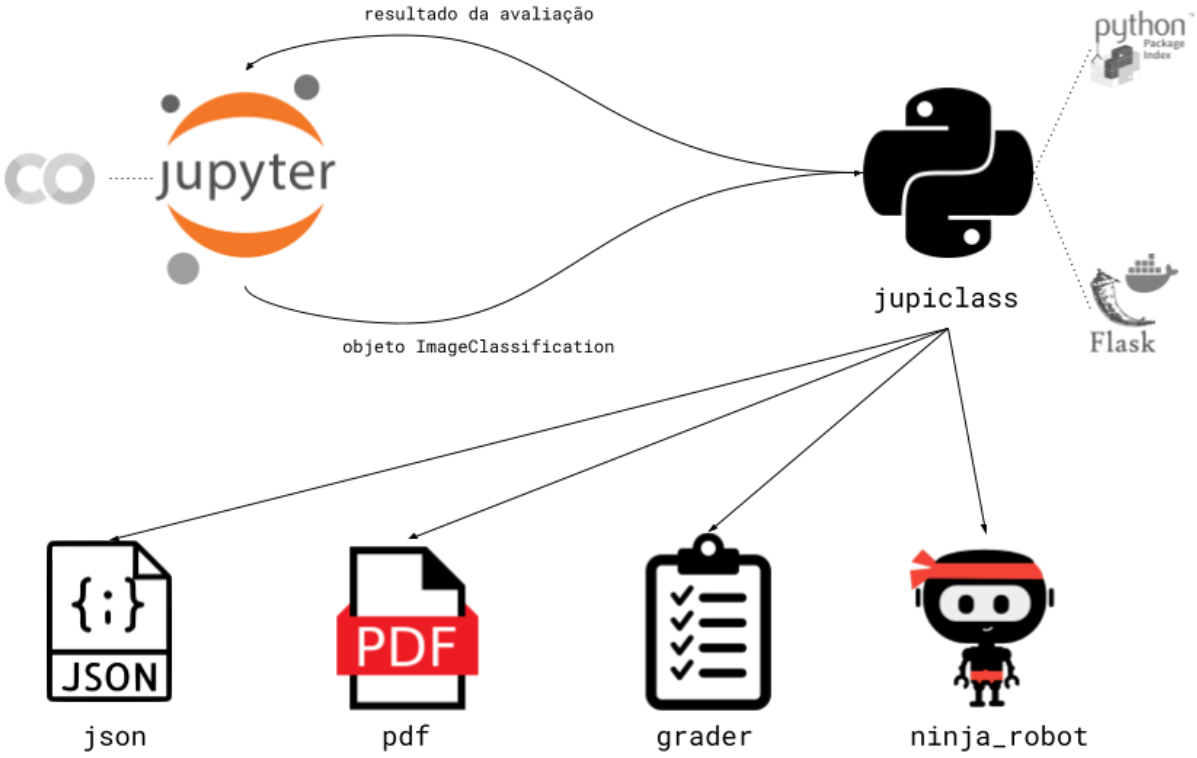


Figura 10 - Arquitetura da automação

5.3 IMPLEMENTAÇÃO

A rubrica de avaliação foi aplicada e testada em um Jupyter Notebook do curso para a classificação de imagens com fast.ai. Algumas informações necessárias do treinamento foram coletadas com *widgets* interativas, como nas

questões de interpretação de acurácia da Figura 11. O arquivo .ipynb com exemplos de coleta das informações está disponibilizado com a biblioteca, podendo servir de base para adaptação e posterior utilização da ferramenta em outra tarefa de classificação de imagens em um Jupyter Notebook.

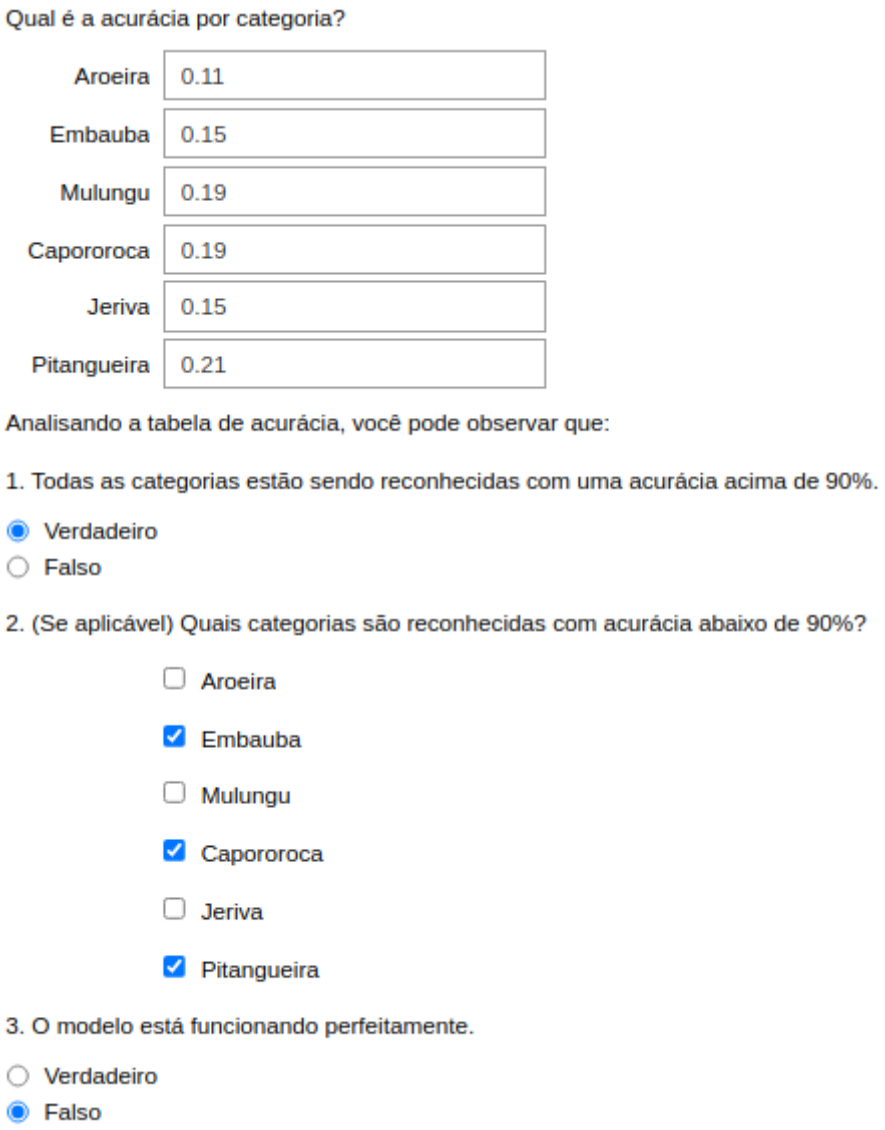


Figura 11 - Widget da questão de interpretação da acurácia

Os critérios da rubrica de avaliação são apresentados a seguir, assim como exemplos dos atributos definidos para o objeto *ImageClassification* com comentários e um pseudocódigo representando o código Python que automatiza a sua correção correspondente.

5.3.1 CRITÉRIOS

CRITÉRIO 1

ID	Critério	Níveis de desempenho		
		Baixo - 0 pt.	***Aceitável - 1 pt.***	***Bom - 2 pt.***
Preparação de dados (OA3)				
C1	Quantidade de imagens	Menos de 5 imagens por categoria	6 de 10 imagens por categoria	Mais de 10 imagens por categoria

```
1.  # Categorias do modelo
2.  # tipo: list[str]
3.  jupic.model_categories = ['Aroeira','Jeriva','Pitangueira','Embauba',
    'Mulungu', 'Capororoca']
4.
5.  # Quantidade de imagens por categoria do dataset
6.  # tipo: list[dict]
7.  jupic.dataset_categories_images = [
8.      { 'Aroeira': 20 },
9.      { 'Capororoca': 20 },
10.     { 'Embauba': 20 },
11.     { 'Jeriva': 20 },
12.     { 'Mulungu': 20 },
13.     { 'Pitangueira': 20 }
14. ]
```

```
avalia_c1():

para cada categoria do modelo
      soma quantidade de imagens em uma pontuação global
            se imagens < 6 -> 0
            se imagens < 10 e imagens > 5 -> 1
            se imagens > 9 -> 2

retorna divisão da pontuação pela quantidade de categorias
```

CRITÉRIO 2

ID	Critério	Níveis de desempenho

		Baixo - 0 pt.	*Aceitável - 1 pt.*	*Bom - 2 pt.*
Preparação de dados (OA3)				
C2	Distribuição do conjunto de dados	Quantidade de imagens por categoria varia muito	Quantidade de imagens por categoria varia pouco	Todas as categorias possuem a mesma quantidade de imagens

```
# Categorias do modelo
# tipo: list[str]
jupic.model_categories = ['Aroeira','Jeriva','Pitangueira','Embauba',
'Mulungu', 'Capororoca']

# Quantidade de imagens por categoria do dataset
# tipo: list[dict]
jupic.dataset_categories_images = [
    { 'Aroeira': 20 },
    { 'Capororoca': 20 },
    { 'Embauba': 20 },
    { 'Jeriva': 20 },
    { 'Mulungu': 20 },
    { 'Pitangueira': 20 }
]
```

```
avalia_c2():

para cada categoria do modelo
      soma número positivo da diferença entre a quantidade de imagens
      da categoria atual e a próxima em uma pontuação global

retorna
      se pontuação > 20 -> 0
      se pontuação > 0 e pontuação < 20 -> 1
      se pontuação == 0 ->: 2
```

CRITÉRIO 3

ID	**Critério**	**Níveis de desempenho**		
		Baixo - 0 pt.	*Aceitável - 1 pt.*	*Bom - 2 pt.*
Preparação de dados/Botânica (OA3/OA12)				
C3	Rotulagem das imagens	Menos de 20% das imagens rotuladas corretamente	De 20% a 99% das imagens rotuladas corretamente	Todas as imagens rotuladas corretamente

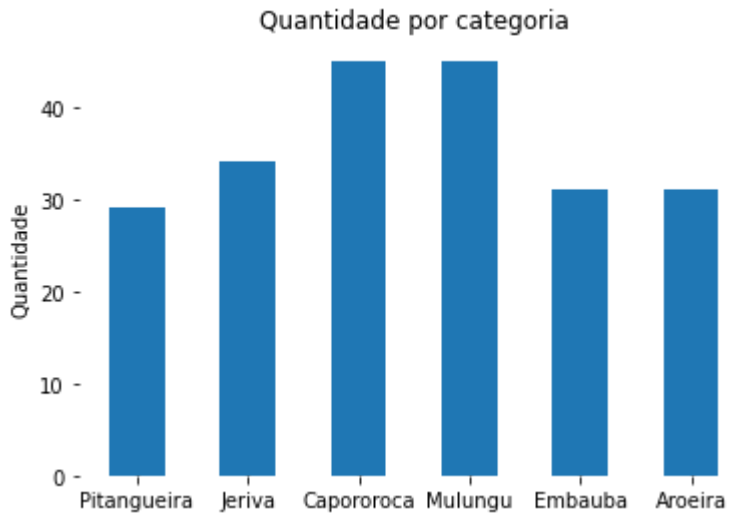
```
# Quantidade de imagens rotuladas corretamente
# tipo: int
jupic.model_correctly_labeled_images = 42

# Quantidade de imagens do dataset
# tipo: int
jupic.dataset_total_images = 100
```

```
avalia_c3():

aplica um modelo de ML de alta precisão para identificar as imagens
rotuladas corretamente

divide as imagens rotuladas corretamente pelo total de imagens no
conjunto de dados

retorna
      se razão < 0.2 -> 0
      se razão > 0.2 e < 0.99 -> 1
      se razão == 1 -> 2
```

O modelo de alta precisão aplicado no curso é documentado conforme apresentado na Tabela 25.

Tabela 25 - Cartão de Modelo do modelo de alta precisão para classificação das árvores

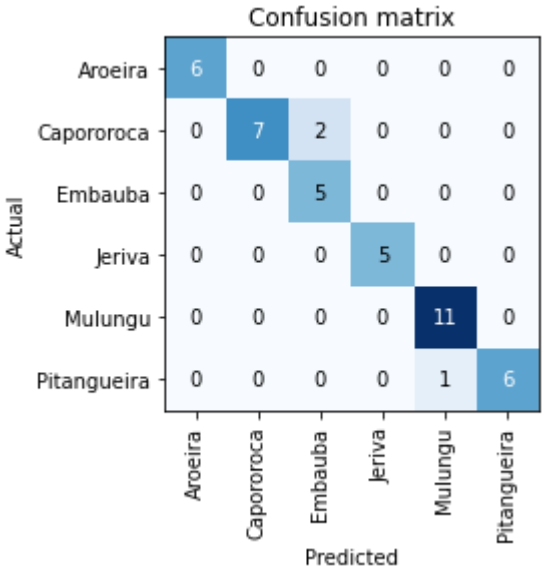
Summary sheet - Modelo de ML	
Nome do modelo	Classificador de Imagens de Árvores com fast.ai
Data	27/07/2021
Versão	v1.0.0
Objetivo do modelo de ML	
Tarefa	Classificar/predizer a espécie de árvore de uma imagem de árvore (tipicamente a vista da árvore toda ou partes dentro do habitat natural (rua, praça, parque, etc.) capturada de um aplicativo Android em relação a 10 categorias de árvores.
Contexto de uso	O modelo é utilizado como exemplo no contexto de ensino de na Educação Básica. Este modelo não foi treinado para ser utilizado em pesquisa na área de botânica.

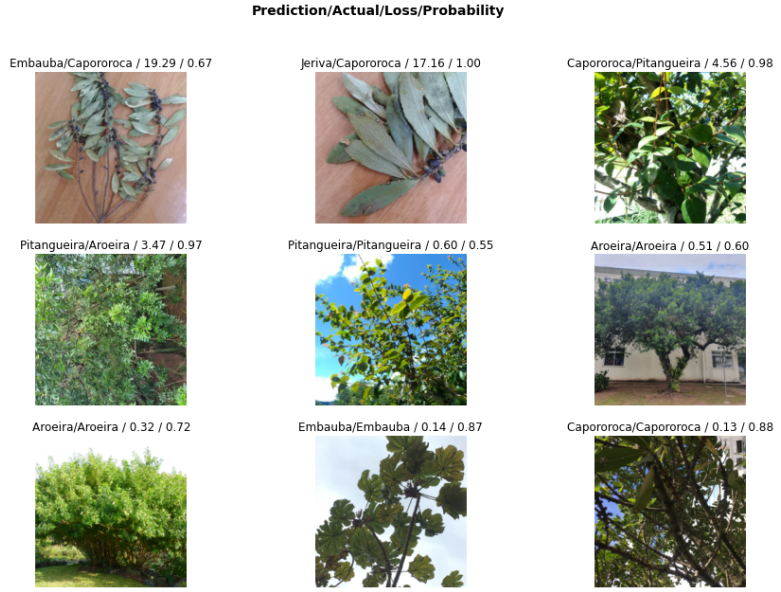
Público alvo	Cidadãos (8+ anos) Foco em alunos do Ensino Médio
Riscos	Risco de classificar erroneamente as espécies de árvores, porém se refere a classificação de árvores sem riscos à saúde dos usuários.
Tipo da tarefa	Single-label classificação de imagens
Categorias	10 categorias de espécies de árvores nativas/endêmicas de SC/Brasil: Aroeira-vermelha, Jerivá, Ipê-amarelo, Ipê-roxo, Mulungu, Capororoca, Embaúba Olandi, Pitangueira, Tanheiro
Conjunto de dados	
Descrição dos dados	Conjunto de imagens de árvores (tipicamente a vista da árvore toda ou partes dentro do habitat natural (rua, praça, parque etc.) capturada de um aplicativo Android
Origem dos dados (coleta própria/uso de conjunto de dados pré-existente, p.ex. do kaggle)	Conjunto de dados de árvores disponibilizado pela CnE Coleta própria via o app “Coleta de imagens CnE”
Quantidade total de dados	Total de 190 imagens
Distribuição dos dados por categoria	
Labeling	Por biólogos (conjunto de dados CnE)
Tipos de aumento de dados aplicados (tipo rotate, crop, etc.)	Resize
Tamanho de imagens	224x224 pixels

Dataset splitting	80% para treinamento (x imagens), 20% para validação (x imagens)
Treinamento - Transfer learning	
Tipo de modelo	ResNet18
Tamanho do batch	Default
Quantidade de épocas	25
Taxa de aprendizagem	5e-3
Avaliação - Transfer learning	
Acurácia por categoria	

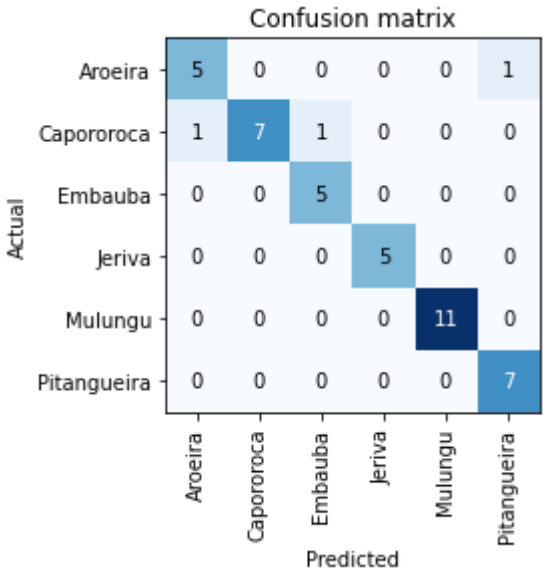
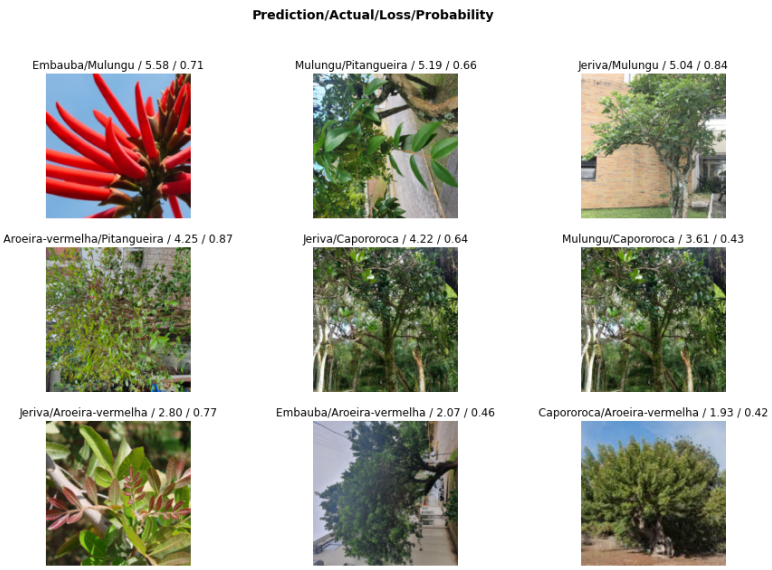
	precision	recall	f1-score	support
Aroeira	1.00	1.00	1.00	6
Capororoca	1.00	0.78	0.88	9
Embauba	0.71	1.00	0.83	5
Jeriva	1.00	1.00	1.00	5
Mulungu	0.92	1.00	0.96	11
Pitangueira	1.00	0.86	0.92	7
accuracy			0.93	43
macro avg	0.94	0.94	0.93	43
weighted avg	0.95	0.93	0.93	43

Matriz de confusão



Top x losses	
Treinamento - Fine-Tuning	
Taxa de aprendizagem	7.585775892948732e-05
Quantidade de épocas	50
Avaliação - Fine-Tuning	
Acurácia por categoria	

	precision	recall	f1-score	support
Aroeira	0.83	0.83	0.83	6
Capororoca	1.00	0.78	0.88	9
Embauba	0.83	1.00	0.91	5
Jeriva	1.00	1.00	1.00	5
Mulungu	1.00	1.00	1.00	11
Pitangueira	0.88	1.00	0.93	7
accuracy			0.93	43
macro avg	0.92	0.94	0.93	43
weighted avg	0.94	0.93	0.93	43

Matriz de confusão	
Top x losses	
Predição	
Quantidade de testes realizados (via upload de imagens novas)	Mais de 1
Limitações e considerações éticas	
Limitações	Esse modelo é limitado a somente 6 espécies nativas de árvores com um desempenho aceitável. Os resultados da classificação devem ser utilizados com cuidado sempre revisado por humanos.
Considerações éticas referente aos dados	Não há considerações éticas referente aos dados.
Referências	

Autores e afiliação	C. Gresse von Wangenheim, R. M. Martins, A. Franz, G. Salvador, INCoD/INE/UFSC

CRITÉRIO 4

ID	Critério	Níveis de desempenho		
		Baixo - 0 pt.	***Aceitável - 1 pt.***	***Bom - 2 pt.***
Preparação de dados/Botânica (OA3/OA12)				
C4	Treinamento - *Transfer Learning*	O modelo não foi treinado (transfer learned)	O modelo foi treinado com os parâmetros padrão	O modelo foi treinado com parâmetros ajustados (arquitetura, época e taxa de aprendizagem)

```
# Modelos utilizados no treinamento
# tipo: list[str]
jupic.tl_models = ["resnet50", "resnet32"]

# Quantidade de épocas utilizadas no treinamento
# tipo: list[int]
jupic.tl_epochs = [1, 2]

# Taxas de aprendizagem utilizadas no treinamento
# tipo: list[float]
jupic.tl_learning_rates = [0.01, 0.002]

# Flag indicando se o modelo foi treinado
# tipo: bool
jupic.tl_trained = True
```

```
avalia_c4():

retorna
      se a flag indicando se modelo foi treinado do modelo for falsa -> 0

      senão
            se um dos arrays de modelos, épocas e taxas de aprendizagem
            tiver tamanho 1 -> 1

            se os arrays de modelos, épocas e taxas de aprendizagem
            tiverem tamanho maior que 1 -> 2
```

CRITÉRIO 5

ID	Critério	Níveis de desempenho		
		Baixo - 0 pt.	***Aceitável - 1 pt.***	***Bom - 2 pt.***
Preparação de dados/Botânica (OA3/OA12)				
C5	Treinamento - *Fine-Tuning*	O modelo não foi fine-tuned	Foi feito *unfreeze* das camadas e melhor taxa de aprendizagem não encontrada ou modelo não treinado	Foi feito *unfreeze* das camadas, a melhor taxa de aprendizagem foi encontrada e o modelo foi *fine-tuned*

```
# Flag indicando se foi feito unfreeze das camadas
# tipo: bool
jupic.ft_unfreezed = True

# Flag indicando se foi a melhor taxa de aprendizagem foi encontrada
# tipo: bool
jupic.ft_learning_rate_found = True

# Flag indicando se o modelo foi fine-tuned
# tipo: bool
jupic.ft_trained = True
```

```
avalia_c5():

retorna
      se a flag de descongelamento da camada for falsa -> 0

      senão

            se a melhor taxa de aprendizagem foi encontrada e o
            modelo foi treinado -> 2

            senão -> 1
```

CRITÉRIO 6

ID	Critério	Níveis de desempenho		
		Baixo - 0 pt.	***Aceitável - 1 pt.***	***Bom - 2 pt.***
Preparação de dados/Botânica (OA3/OA12)				

C6	Interpretação de acurácia	Categorias com baixa acurácia não identificadas	Categorias com baixa acurácia identificadas e interpretação incorreta em relação ao modelo	Categorias com baixa acurácia identificadas corretamente e interpretação correta em relação ao modelo

```python
# Resultado da pergunta de análise de acurácia (Transfer Learning)
# tipo: str
jupic.tl_accuracy_analysis = True

# Categorias com acurácia abaixo de 90% (Transfer Learning)
# tipo: list[str]
jupic.tl_accuracy_analysis_categories = ["Aroeira", "Capororoca", "Embauba", "Jeriva", "Mulungu", "Pitangueira"]

# Resultado da pergunta de interpretação de acurácia (Transfer Learning)
# tipo: bool
jupic.tl_accuracy_interpretation = False

# Resultado da pergunta de análise de acurácia (Fine-Tuning)
# tipo: str
jupic.ft_accuracy_analysis = True

# Categorias com acurácia abaixo de 90% (Fine-Tuning)
# tipo: list[str]
jupic.ft_accuracy_analysis_categories = ["Aroeira", "Capororoca", "Embauba", "Jeriva", "Mulungu", "Pitangueira"]

# Resultado da pergunta de interpretação de acurácia (Fine-Tuning)
# tipo: bool
jupic.ft_accuracy_interpretation = False
```

```
avalia_c6():

para Transfer Learning e Fine-Tuning, pega o resultado do abaixo e divide por 2:

pede para estudante preencher as acurácias por categoria
pergunta se todas as categorias possuem acurácia acima de 90%

se estudante diz que sim
      se todas as categorias possuem acurácia acima de 90%
            retorna
                  se estudante diz que modelo está funcionando -> 2
                  se estudante diz que modelo não está funcionando -> 1
senão
      retorna
            se nem todas as categorias possuem acurácia acima de 90%
```

```
                se estudante diz que modelo está funcionando -> 0

                se categorias com baixa acurácia identificadas
                      se estudante diz que modelo está
                      funcionando -> 1

                      se estudante diz que modelo não está
                      funcionando -> 2
```

CRITÉRIO 7

ID	Critério	Níveis de desempenho		
		Baixo - 0 pt.	***Aceitável - 1 pt.***	***Bom - 2 pt.***
Preparação de dados/Botânica (OA3/OA12)				
C7	Interpretação da matriz de confusão	Classificações incorretas não identificadas e interpretação incorreta em relação ao modelo	Classificações incorretas identificadas e interpretação incorreta em relação ao modelo	Classificações incorretas identificadas e interpretação correta em relação ao modelo

```
1. # Dicionário de categorias e quais categorias identificadas incorretamente,
   obtidas da matriz de confusão (Transfer Learning)
2. # tipo: list[dict]
3. jupic.tl_confusion_matrix_mislabeled_real = [{'Pitangueira': ['Capororoca',
   'Aroeira', 'Embauba', 'Mulungu']}, {'Jeriva': []}, {'Capororoca':
   ['Jeriva', 'Aroeira']}, {'Mulungu': []}, {'Embauba': []}, {'Aroeira':
   ['Pitangueira', 'Embauba', 'Jeriva', 'Mulungu']}]
4.
5. # Dicionário de categorias e quais categorias identificadas incorretamente,
   resposta do estudante (Transfer Learning)
6. # tipo: list[dict]
7. jupic.tl_confusion_matrix_mislabeled = [{'Pitangueira': ['Aroeira',
   'Capororoca', 'Embauba', 'Mulungu']}, {'Jeriva': []}, {'Capororoca':
   ['Aroeira', 'Jeriva']}, {'Mulungu': []}, {'Embauba': []}, {'Aroeira':
   ['Embauba', 'Jeriva', 'Mulungu', 'Pitangueira']}]
8.
9. # Resultado da pergunta de análise de matriz de confusão (Transfer
   Learning)
10. # tipo: bool
11. jupic.tl_confusion_matrix_interpretation = False
12.
13. # Dicionário de categorias e quais categorias identificadas incorretamente,
   obtidas da matriz de confusão (Fine-Tuning)
14. # tipo: list[dict]
15. jupic.ft_confusion_matrix_mislabeled_real = [{'Pitangueira': ['Capororoca',
   'Aroeira', 'Embauba', 'Mulungu']}, {'Jeriva': []}, {'Capororoca':
```

```
    ['Jeriva', 'Aroeira']}, {'Mulungu': []}, {'Embauba': []}, {'Aroeira':
    ['Pitangueira', 'Embauba', 'Jeriva', 'Mulungu']}]
16.
17. # Dicionário de categorias e quais categorias identificadas incorretamente,
    resposta do estudante (Fine-Tuning)
18. # tipo: list[dict]
19. jupic.ft_confusion_matrix_mislabeled = [{'Pitangueira': ['Aroeira',
    'Caporoca', 'Embauba', 'Mulungu']}, {'Jeriva': []}, {'Caporoca':
    ['Aroeira', 'Jeriva']}, {'Mulungu': []}, {'Embauba': []}, {'Aroeira':
    ['Embauba', 'Jeriva', 'Mulungu', 'Pitangueira']}]
20.
21. # Resultado da pergunta de análise de matriz de confusão (Fine-Tuning)
22. # tipo: bool
23. jupic.ft_confusion_matrix_interpretation = False
```

```
avalia_c7():

para Transfer Learning e Fine-Tuning, pega o resultado do abaixo e divide
por 2:

pede para estudante dizer quais categorias foram erroneamente rotuladas

compara as categorias da resposta com as categorias da matriz de confusão

pergunta está o modelo está funcionando perfeitamente
       se identificou corretamente as classificações:
             retorna:
                   se diz que sim:
                         se não há erro de rotulagem -> 2
                         se há erros -> 1
                   senão:
                         se há erros de rotulagem -> 2
                         se há erros -> 1
```

CRITÉRIO 8

ID	Critério	Níveis de desempenho		
		Baixo - 0 pt.	***Aceitável - 1 pt.***	***Bom - 2 pt.***
Preparação de dados/Botânica (OA3/OA12)				
C8	Ajustes/melhorias feitas	Sem novas iterações de desenvolvimento	Uma nova iteração com alterações no conjunto de dados e/ou parâmetros de treinamento	Diversas novas iterações com alterações no conjunto de dados e/ou parâmetros de

				treinamento

```
# Resposta de quantas iterações foram feitas
# tipo: int
jupic.performance_tuning = 0

# Resposta de quais alterações foram feitas e se o desempenho do modelo melhorou
# tipo: str
jupic.performance_tuning_text = "Acurácia, melhorou"
```

```
avalia_c8():

pede para estudante dizer se houve tentativas de melhorias

retorna
      se houve tentativa de melhoria
            há comentário sobre o que foi alterado -> 2
            senão -> 1
      senão -> 0
```

CRITÉRIO 9

ID	Critério	Níveis de desempenho		
		Baixo - 0 pt.	***Aceitável - 1 pt.***	***Bom - 2 pt.***
Avaliação e interpretação do desempenho de um modelo de ML/Testar e aperfeiçoar programas (OA5/OA11)				
C9	Testes com novos objetos	Nenhum novo objeto testado	1-2 novos objetos testados	Mais de dois novos objetos testados

```
# Categorias rotuladas para os novos objetos testados
# tipo: list[str]
jupic.predicted_objects = ["Mulungu", "Jeriva", "Embauba", "Pitangueira", "Mulungu"]
```

```
avalia_c9():

pede novos objetos testados ao estudante

retorna
```

```
      se objetos testados > 2 -> 2
      se objetos testados < 3 e objetos testados > 0 -> 2
      se objetos testados == 0 -> 0
```

CRITÉRIO 10

ID	Critério	Níveis de desempenho		
		Baixo - 0 pt.	***Aceitável - 1 pt.***	***Bom - 2 pt.***
Avaliação e interpretação do desempenho de um modelo de ML/Testar e aperfeiçoar programas (OA5/OA11)				
C10	Interpretação dos testes	Interpretação errada	---	Interpretação correta

```python
# Categorias reais para os novos objetos testados
# tipo: list[str]
jupic.real_objects = ["Embauba", "Embauba", "Jeriva", "Jeriva",
"Pitangueira"]

# Categorias rotuladas para os novos objetos testados
# tipo: list[str]
jupic.predicted_objects = ["Mulungu", "Jeriva", "Embauba", "Pitangueira",
"Mulungu"]

# Quantidade de vezes que o modelo acertou
# tipo: int
jupic.predicted_success_times = 5

# Resposta da questão de interpretação sobre os novos testes
# tipo: bool
jupic.predicted_success_interpretation = True
```

```
avalia_c10():

pede novos objetos testados e a predição do modelo ao estudante
pede quantidade de acertos do modelo ao estudante
pede se o modelo está funcionando corretamente ao estudante

itera nos arrays de objetos preditos e objetos testados, comparando
os itens por posição. se iguais, soma em uma pontuação global

retorna
      se pontuação == quantidade de acertos
            se pontuação == quantidade de objetos e estudante diz que
            o modelo está funcionando -> 2
```

```
        senão
              se estudante diz que o modelo não está
              funcionando -> 2
```

5.3.2 USO DA AUTOMAÇÃO

O processo de instalação da biblioteca em um Jupyter Notebook e um exemplo resumido de uso são apresentados a seguir.

```
# Instala o pacote
pip install jupiclass

# Importa as funções
from jupiclass import ImageClassification, evaluate, write_json,
get_ninja_robot, PDFWriter

# Cria um novo objeto ImageClassification
jupic = ImageClassification()

# Defina os atributos do ImageClassification...

# Avalia o treinamento da classificação de imagens
score = evaluate(jupic)

# Salva JSON com as informações do treinamento
write_json(jupic)

# Salva imagem do ninja robô em './ninja_robot.png'
get_ninja_robot(score['total_score'])

# Salva PDF com as informações do treinamento e questões de interpretação
# recebendo como parâmetros o caminho da imagem da logo, distribuição de
# dados, matriz de confusão e top losses do Transfer Learning e Fine-Tuning
writer = PDFWriter('logo.png', 'dataset_distribution.png',
    'tl_confusion_matrix.png', 'tl_top_losses.png',
    'ft_confusion_matrix.png', 'ft_top_losses.png')
writer.write(jupic)
```

5.3.3 EXEMPLO DE OBJETO IMAGECLASSIFICATION

```
jupic = ImageClassification()
```

```
##### Model
jupic.model_categories = ['Aroeira','Jeriva','Pitangueira','Embauba',
'Mulungu', 'Capororoca']
jupic.model_correctly_labeled_images = 100
###### Dataset
jupic.dataset_categories_images = [{ 'Aroeira': 20 }, { 'Capororoca': 20 },
{ 'Embauba': 20 }, { 'Jeriva': 20 }, { 'Mulungu': 20 }, { 'Pitangueira': 20
}]
jupic.dataset_total_images = 100
###### Transfer Learning
jupic.tl_models = ['resnet50', 'resnet32']
jupic.tl_epochs = [1, 2]
jupic.tl_learning_rates = [0e-1, 0e-2]
jupic.tl_trained = True
##### Accuracy - Transfer Learning
jupic.tl_accuracy_categories = [{ 'Aroeira': 0.9 }, { 'Capororoca': 0.9 },
{ 'Embauba': 0.9 }, { 'Jeriva': 0.9 }, { 'Mulungu': 0.9 }, { 'Pitangueira':
0.9 }]
jupic.tl_accuracy_analysis = 'Verdadeiro'
jupic.tl_accuracy_analysis_categories = ['Aroeira', 'Capororoca',
'Embauba', 'Jeriva', 'Mulungu', 'Pitangueira']
jupic.tl_accuracy_interpretation = 'Falso'
##### Confusion matrix- Transfer Learning
jupic.tl_confusion_matrix_mislabeled_real = [{'Pitangueira': ['Capororoca',
'Aroeira', 'Embauba', 'Mulungu']}, {'Jeriva': []}, {'Capororoca':
['Jeriva', 'Aroeira']}, {'Mulungu': []}, {'Embauba': []}, {'Aroeira':
['Pitangueira', 'Embauba', 'Jeriva', 'Mulungu']}]
jupic.tl_confusion_matrix_mislabeled = [{'Pitangueira': ['Aroeira',
'Capororoca', 'Embauba', 'Mulungu']}, {'Jeriva': []}, {'Capororoca':
['Aroeira', 'Jeriva']}, {'Mulungu': []}, {'Embauba': []}, {'Aroeira':
['Embauba', 'Jeriva', 'Mulungu', 'Pitangueira']}]
jupic.tl_confusion_matrix_interpretation = 'Falso'
##### Fine-Tuning
jupic.ft_unfreezed = True
jupic.ft_learning_rate_found = True
jupic.ft_trained = True
##### Accuracy - Fine-Tuning
jupic.ft_accuracy_categories = [{ 'Aroeira': 0.1 }, { 'Capororoca': 0.2 },
{ 'Embauba': 0.3 }, { 'Jeriva': 0.4 }, { 'Mulungu': 0.5 }, { 'Pitangueira':
0.6 }]
jupic.ft_accuracy_analysis = 'Falso'
jupic.ft_accuracy_analysis_categories = ['Aroeira', 'Capororoca',
'Embauba', 'Jeriva', 'Mulungu', 'Pitangueira']
jupic.ft_accuracy_interpretation = 'Falso'
##### Confusion matrix - Fine-Tuning
jupic.ft_confusion_matrix_mislabeled_real = [{'Pitangueira': ['Capororoca',
'Aroeira', 'Embauba', 'Mulungu']}, {'Jeriva': []}, {'Capororoca':
['Jeriva', 'Aroeira']}, {'Mulungu': []}, {'Embauba': []}, {'Aroeira':
['Pitangueira', 'Embauba', 'Jeriva', 'Mulungu']}]
jupic.ft_confusion_matrix_mislabeled = [{'Pitangueira': ['Capororoca',
'Aroeira', 'Embauba', 'Mulungu']}, {'Jeriva': []}, {'Capororoca':
['Jeriva', 'Aroeira']}, {'Mulungu': []}, {'Embauba': []}, {'Aroeira':
['Pitangueira', 'Embauba', 'Jeriva', 'Mulungu']}]
jupic.ft_confusion_matrix_interpretation = 'Falso'
##### Performance
jupic.performance_tuning = 2
```

```
37. jupic.performance_tuning_text = 'Acurácia'
38. ##### New objects
39. jupic.real_objects = ['Aroeira', 'Aroeira', 'Aroeira', 'Aroeira',
    'Aroeira']
40. jupic.predicted_objects = ['Aroeira', 'Aroeira', 'Aroeira', 'Aroeira',
    'Aroeira']
41. jupic.predicted_success_times = 5
42. jupic.predicted_success_interpretation = 'Verdadeiro'
```

5.3.4 APRESENTAÇÃO DA AVALIAÇÃO

Os resultados da avaliação são apresentados no Jupyter Notebook utilizando ipywidgets. Como resultado é apresentada a pontuação obtida para cada um dos critérios da rubrica (Tabela 20). É apresentada também a nota final calculada a partir das pontuações conforme definido no Modelo de Medição.

Junto com a pontuação do treinamento do modelo é apresentado um mascote “ninja-robô” representando a nota de forma lúdica, incentivando o engajamento dos estudantes para melhorar seu resultado e obter novas faixas. Os “ninja-robôs” possíveis de alcançar são apresentados na Tabela 26.

Tabela 26 - Escala de “ninjas-robôs” conforme a pontuação

										
0	1	2	3	4	5	6	7	8	9	10

Um exemplo de apresentação para um treinamento com pontuação 5 e faixa verde é apresentado na Figura 12, usando *ipywidgets* para apresentar visualmente o dicionário com a pontuação por critério.

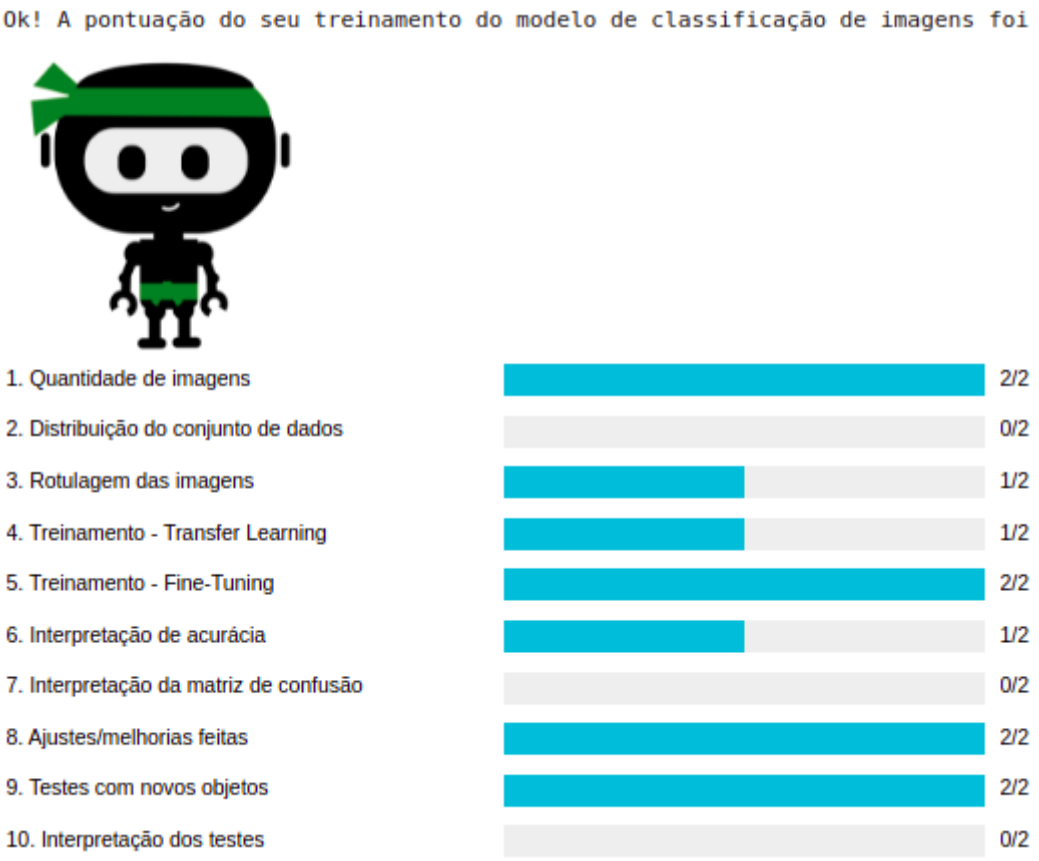


Figura 12 - Exemplo de apresentação da avaliação para a nota 5

5.3.5 GERAÇÃO DO RELATÓRIO DO MODELO TREINADO

O pacote disponibiliza um módulo para efetuar a construção de um Modelo de Documentação com as informações do modelo em formato PDF. Para a construção do PDF, é necessário setar alguns atributos extras no objeto ImageClassification.

```
1. jupic.model_name = 'Modelo Teste'
2. jupic.model_date = '26/08/2021'
3. jupic.model_version = 'v1.0.0'
4. jupic.model_task = 'Classificar a espécie de árvore de uma imagem de árvore
   capturada de um aplicativo Android em relação a 6 categorias de árvores
   nativas/endêmicas de SC/Brasil. '
5. jupic.model_use_context = 'O modelo é utilizado como exemplo no contexto de
   ensino de na Educação Básica. Este modelo não foi treinado para ser
   utilizado em pesquisa na área de botânica.'
6. jupic.model_target_audience = 'Cidadãos (8+ anos), Foco em alunos do Ensino
   Médio'
```

```
jupic.model_risks = 'Risco de classificar erroneamente as espécies de árvores, porém se refere a classificação de árvores sem riscos à saúde dos usuários.'
jupic.model_task_type = 'Single-label classificação de imagens'
jupic.dataset_augmentation_size = '224x224 pixels'
jupic.dataset_augmentation_type = 'rotate'
jupic.dataset_description = 'Conjunto de imagens de árvores (tipicamente a vista da árvore toda ou partes dentro do habitat natural (rua, praça, parque etc.) capturada de um aplicativo Android'
jupic.dataset_origin = 'Conjunto de dados de árvores disponibilizado pela CnE'
jupic.dataset_validation_percentage = 0.2
jupic.dataset_labeler_name = 'Por biólogos (conjunto de dados CnE)'
jupic.tl_batch_size = '200'
jupic.ft_learning_rate = '0.005'
jupic.ft_epoch = '35'
jupic.ethics_limitations = 'Esse modelo é limitado a somente 6 espécies nativas de árvores com um desempenho aceitável. Os resultados da classificação devem ser utilizados com cuidado sempre revisado por humanos.'
jupic.ethics_considerations = 'N/A'
jupic.author = 'C. Gresse von Wangenheim, R. M. Martins, A. Franz, G. Salvador, INCoD/INE/UFSC'
```

5.3.6 DISPONIBILIZAÇÃO DA AUTOMAÇÃO

A automação é disponibilizada via pacote no PyPI ou via servidor HTTP em Flask, disponível como container Docker. O pacote possui o nome de jupiclass e está disponível conforme apresentado em http://computacaonaescola.paginas.ufsc.br/jupiclass/. Para a publicação foi utilizada a biblioteca Poetry (2021), simplificando o processo de geração de versões e upload do pacote no PyPI, utilizando os comandos o arquivo *pyproject.toml* apresentado a seguir.

```
# Gera versão, podendo ser 'patch', 'minor' ou 'major'
poetry version patch

# Publica pacote no PyPI
poetry publish --build
```

```
[build-system]
requires = [
    "setuptools>=42",
    "wheel"
]
build-backend = "setuptools.build_meta"

[tool.poetry]
name = "jupiclass"
version = "1.0.0"
description = "Jupyter Notebook Image Classification assessment tool"
license = "BSD-3-Clause"
authors = [
    "gustavo.castro.salvador <gustavo.castro.salvador@grad.ufsc.br>"
]
readme = 'README.md'
repository = "https://codigos.ufsc.br/gqs/jupiclass"
homepage = "https://codigos.ufsc.br/gqs/jupiclass"

[tool.poetry.dependencies]
python = "~2.7 || ^3.2"
fpdf = "*"
PyPDF2 = "*"
```

A avaliação também é disponibilizada via servidor HTTP, com uma API utilizando Flask. A API é executada via Docker, podendo ser gerada no repositório com o *Dockerfile* e *requirements.txt* a seguir.

```
FROM python:3.8-slim-buster

WORKDIR /app

COPY requirements.txt requirements.txt
RUN pip3 install -r requirements.txt

COPY . .

CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```

```
Flask==2.0.1
```

A API possui apenas uma rota, **`POST /api/evaluate`**. Ela aceita um JSON como corpo da requisição. Este JSON representa o objeto *ImageClassification* preenchido, podendo ser gerado com o pacote *json* ou apenas enviando o objeto diretamente. Um exemplo do JSON pode ser encontrado no Apêndice 1.

O resultado é retornado como um JSON contendo as competências e respectivas notas, além de uma nota total.

```
{
    'category_images': 2,                   # C1
    'dataset_distribution': 2,              # C2
    'model_predictions': 2,                 # C3
    'transfer_learning': 2,                 # C4
    'fine_tuning': 2,                       # C5
    'accuracy': 1,                          # C6
    'confusion_matrix': 1,                  # C7
    'performance_tuning': 2,                # C8
    'new_tests': 2,                         # C9
    'new_tests_interpretation': 2,          # C10
    'total_score': 9                        # soma dos critérios/2
}
```

Os arquivos do projeto podem ser encontrados no repositório GitLab institucional da Universidade Federal de Santa Catarina, em https://codigos.ufsc.br/gqs/jupiclass.

5.3.7 TUTORIAL DE UTILIZAÇÃO

Em resumo, para a utilização do *jupiclass* em um Jupyter Notebook:

1. Instalar a biblioteca do PyPI, importar a classe *ImageClassification* e as funções desejadas:

```
pip install jupiclass

from jupiclass import ImageClassification, evaluate, write_json,
get_ninja_robot, PDFWriter
```

2. Criar um objeto da classe *ImageClassification* e preencher seus atributos seguindo o exemplo da seção 5.3.2. Alguns atributos devem ser preenchidos apenas para a geração do PDF, como na seção 5.3.5.

3. Utilizar as funções disponíveis, conforme seção 5.3.3. Os arquivos (PDF e JSON) são salvos no mesmo diretório em que a ferramenta é utilizada.

6 AVALIAÇÃO DA AUTOMAÇÃO

6.1 DEFINIÇÃO DA AVALIAÇÃO

Para avaliar a qualidade do modelo de avaliação, foi realizada uma avaliação preliminar com o objetivo de analisar a qualidade da ferramenta desenvolvida em termos de utilidade, adequação funcional, eficiência de desempenho e usabilidade do ponto de vista de professores e alunos no contexto do ensino de *Machine Learning*. Detalhes da avaliação podem ser encontrados no Apêndice 2.

Com base na ISO/IEC 25010 (2011), ISO/IEC 9241 (1998), TAM (Davis, 1989) e SUS (BROOKE, 1996), os fatores de qualidade a serem avaliados são decompostos (Tabela 27).

Tabela 27 - Visão geral da decomposição das características de qualidade e operacionalização da medição

Característica	Subcaracterística	Item no questionário	Escala de resposta
Utilidade	--	Acho a ferramenta de avaliação útil na educação de computação no ensino fundamental e médio.	Escala Likert de 5 pontos (Concordo totalmente, Concordo, Não concordo nem discordo; Discordo; Discordo totalmente)
		Acho a ferramenta de avaliação útil no ensino de computação para iniciantes em cursos de graduação.	
		Acho a ferramenta de avaliação útil na educação de computação para iniciantes em cursos de pós-graduação.	
		Acho que em sua forma atual a ferramenta de avaliação pode ser aplicada de forma prática em minhas aulas.	
Adequação funcional	Completude funcional	Você acha que existem aspectos/critérios para avaliar a construção de modelos de ML para Classificação de Imagens no contexto de ensino de ML no ensino médio que não são suportados pela ferramenta?	Sim, não (se sim, qual)
		Você acha que existem aspectos relevantes no que diz respeito ao processo de avaliação da construção de modelos de ML para Classificação de Imagens no contexto de ensino de ML no ensino médio que não são apoiados pela ferramenta de avaliação?	
		Você acha que as informações de feedback fornecidas são suficientes?	Sim, não (se não, o que está faltando)
	Corretude funcional	Você notou algum erro em relação à funcionalidade da ferramenta de avaliação?	Sim, não (se sim, qual)
		Você achou os resultados da avaliação corretos?	Sim, não (se não, explique)
Eficiência de desempenho	Comportamento de tempo	O desempenho em termos de tempo de processamento da ferramenta de avaliação é aceitável (não leva muito tempo)?	Sim, não
Usabilidade	Eficácia	Você conseguiu concluir uma avaliação usando a ferramenta de avaliação?	
	Eficiência	Você considera que o tempo que leva para interagir com a ferramenta de avaliação para obter feedback é adequado (não leva muito tempo)?	
		Comparando o esforço/tempo que você levou para avaliar o projeto de ML dos alunos com uma avaliação manual, você acha que isso reduzirá sua carga de trabalho?	
	Satisfação	Acho que gostaria de usar este sistema com frequência.	Escala Likert de 5 pontos (Concordo totalmente, Concordo, Não concordo nem discordo; Discordo; Discordo totalmente)
		Achei o sistema desnecessariamente complexo.	
		Achei o sistema fácil de usar.	
		Acho que precisaria do apoio de um técnico para poder usar este sistema.	
		Achei que as várias funções deste sistema estavam bem integradas.	
		Achei que havia muita inconsistência neste sistema.	
		Eu imagino que a maioria das pessoas aprenderia a usar esse sistema muito rapidamente.	
		Achei o sistema muito complicado de usar.	
		Eu me senti muito confiante ao usar o sistema.	
		Eu precisava aprender muitas coisas antes de começar a usar este sistema.	
	Operabilidade	Você acha que a ferramenta de avaliação possui elementos ambíguos ou difíceis de entender?	Sim, não (se sim, qual)
Pontos fortes e fracos	--	O que você mais gostou na ferramenta de avaliação?	Aberto (texto longo)
		O que você menos gostou na ferramenta de avaliação?	Aberto (texto longo)
		Mais alguma sugestão?	Aberto (texto longo)

A avaliação foi realizada por meio de um teste de usuário com um painel de especialistas. O teste visa avaliar a qualidade percebida do ponto de vista de professores e alunos. Durante o teste do usuário, os usuários recebem primeiro uma visão geral básica sobre o objetivo e os recursos da ferramenta de avaliação. Em seguida, eles realizam uma tarefa predefinida (executar e avaliar um projeto de ML para a classificação de imagens de árvores nativas com a ferramenta).

Os dados foram coletados por meio de um questionário online utilizando LimeSurvey (LIMESURVEY, 2021) coletando dados demográficos no início do teste e dados de avaliação da biblioteca no final do teste. Os itens do questionário foram derivados das características de qualidade (Tabela 27). Foi utilizada uma escala nominal para a maioria dos itens do questionário (sim/não), com exceção dos itens de satisfação. Essa característica de qualidade é medida pela adoção do questionário SUS (BROOKE, 1996) com uma escala Likert de 5 pontos. Além disso, também foi solicitado aos participantes que identificassem pontos fortes e fracos da ferramenta.

6.2 EXECUÇÃO DA AVALIAÇÃO

A avaliação da ferramenta foi realizada em Agosto de 2021 por um total de 14 participantes, incluindo professores do Ensino Médio e ensino superior e alunos do Ensino Superior em Santa Catarina/Brasil conforme detalhado na Figura 13. Nesta avaliação inicial, devido à atual falta de competências em ML entre os alunos do Ensino Médio, foram convidados alunos de graduação e pós-graduação para o estudo.

Foram convidados no total 23 participantes via e-mail, dos quais 14 responderam a pesquisa representando uma taxa de resposta de 60%. Destas 14 respostas, foram 7 completas. Ocorreram problemas técnicos durante a execução do questionário decorrentes da execução simultânea de Jupyter Notebooks em sessões diferentes do Google Colab, que podem ter afetado a desistência de completar o questionário. Esses problemas poderiam ser resolvidos se os

participantes executassem o arquivo do Notebook (.ipynb) em contas separadas, compartilhando o conjunto de dados de um Google Drive único.

Figura 13 - Área de atuação dos participantes (os participantes poderiam indicar mais de uma área de atuação)

Em termos de competências de ML os participantes variam em relação à questão de experiência prévia com desenvolvimento de modelos de ML para classificação de imagens com Jupyter Notebook (Figura 14). A maioria dos participantes já havia desenvolvido mais de 3 modelos.

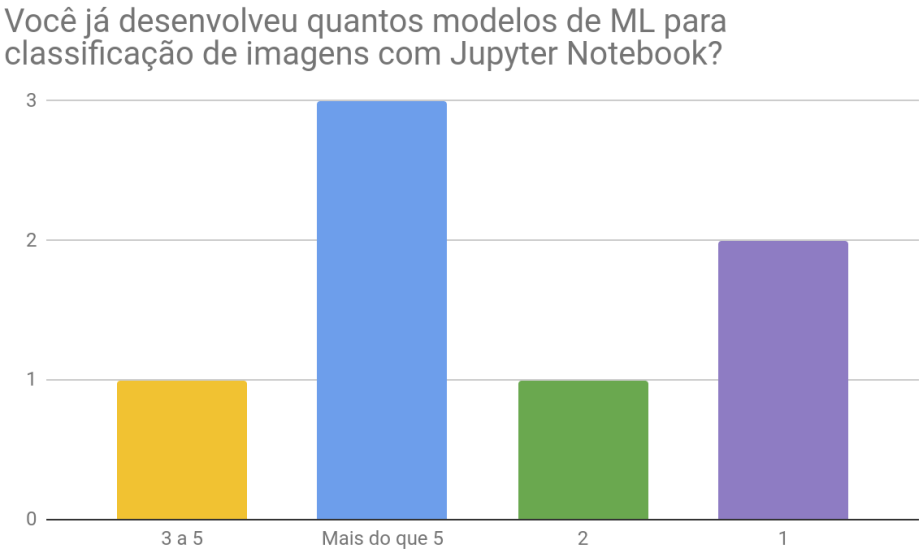


Figura 14 - Resultado da questão de experiência prévia com desenvolvimento de modelos com Jupyter

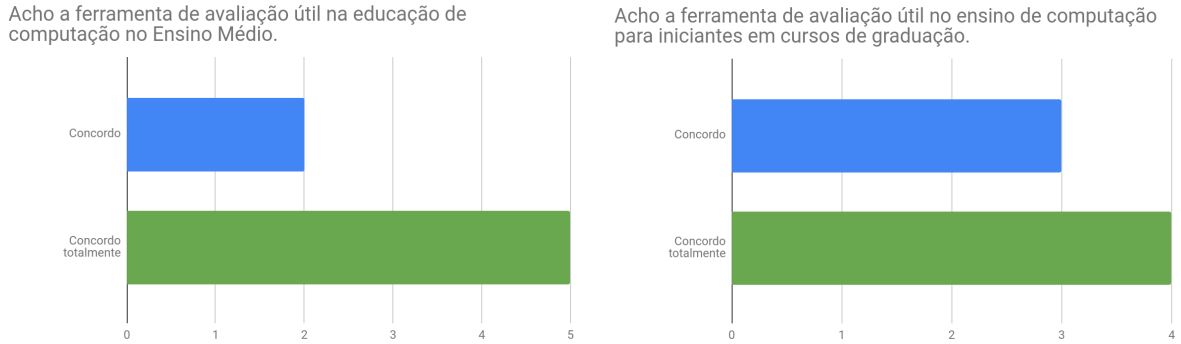
Os professores participantes da avaliação relataram que não ensinam *Machine Learning* em suas disciplinas. Os dados coletados são apresentados no Apêndice 3.

6.3 ANÁLISE DA AVALIAÇÃO

De acordo com os fatores de qualidade definidos (Tabela 27), os dados coletados foram analisados.

O JUPICLASS É ÚTIL?

Todos os participantes consideraram a ferramenta de avaliação útil no ensino de ML no Ensino Médio, assim como para o ensino de iniciantes em cursos de graduação e pós-graduação. Além disso, a maioria dos professores consideraram que a ferramenta em seu estado atual pode ser aplicada de forma prática em suas aulas. Os resultados são apresentados na Figura 15.



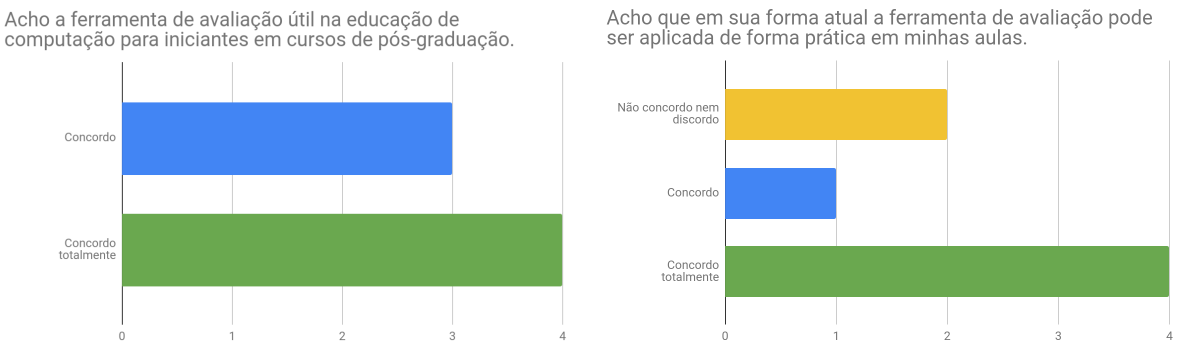


Figura 15 - Resultados das questões de utilidade da ferramenta

O JUPICLASS É ADEQUADO FUNCIONALMENTE?

Todos os participantes indicaram que a ferramenta suporta todos os aspectos/critérios para avaliar a construção de modelos de ML para classificação de imagens no contexto de ensino de ML no Ensino Médio, assim como aspectos relevantes no que diz respeito ao processo de avaliação. Dois participantes fizeram uma sugestão de incluir quais foram os erros cometidos pelo estudante durante o treinamento do modelo nas informações de *feedback* apresentadas.

A maioria dos participantes se deparou com erros durante a execução do Colab, decorrentes dos problemas técnicos já relatados de execução simultânea, além de problemas com GPUs do Google Colab e com o *widget* da questão de interpretação da matriz de confusão. Mensagens de erro mais explicativas poderiam ter guiado melhor os participantes na execução. Embora a maioria dos participantes achou o resultado da avaliação correto, alguns questionaram como o cálculo das questões de interpretação é feito, dado que os critérios avaliados e suas respectivas pontuações não foram apresentados em um primeiro momento.

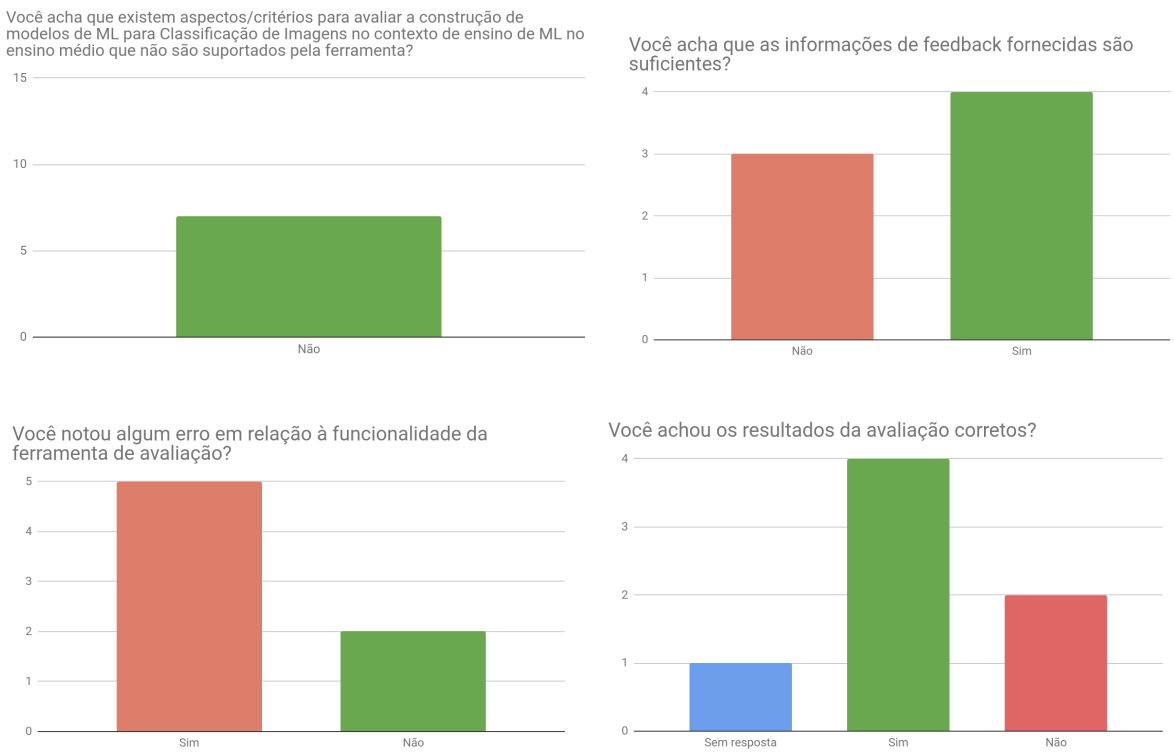


Figura 16 - Resultados das questões de adequação funcional da ferramenta

O JUPICLASS É EFICIENTE EM TERMOS DE DESEMPENHO?

Quase todos os participantes consideraram aceitável o desempenho da ferramenta de avaliação em termos de tempo de processamento.

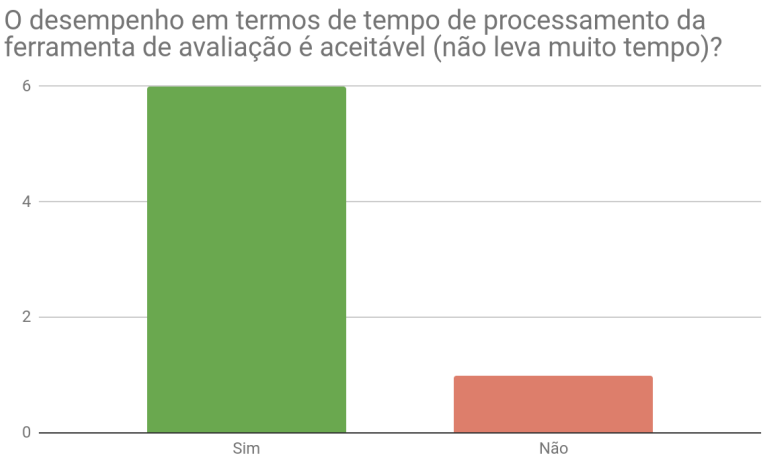


Figura 17 - Resultado da questão de desempenho da ferramenta

O JUPICLASS POSSUI BOA USABILIDADE?

Todos os participantes que responderam completamente o questionário conseguiram concluir ao final uma avaliação do treinamento do modelo no Google Colab com a ferramenta de avaliação.

Eles consideram que o tempo que leva para interagir com a ferramenta de avaliação para obter *feedback* é adequado. Todos acreditam que a ferramenta reduz a carga de trabalho em comparação com uma avaliação manual de projeto de ML.

Aplicando o *System Usability Scale* (SUS) (BROOKE, 1996) para medir a satisfação com a ferramenta, a pontuação média foi de 81.78, indicando um nível de usabilidade aceitável e satisfação muito boa.

Tabela 28 - Resultados da pontuação SUS

	Média	**Valor mínimo**	**Valor máximo**
jupiclass	81.78	65	92.5

O entendimento dos participantes da existência de elementos ambíguos ou difíceis de entender na ferramenta é apresentado na Figura 18.

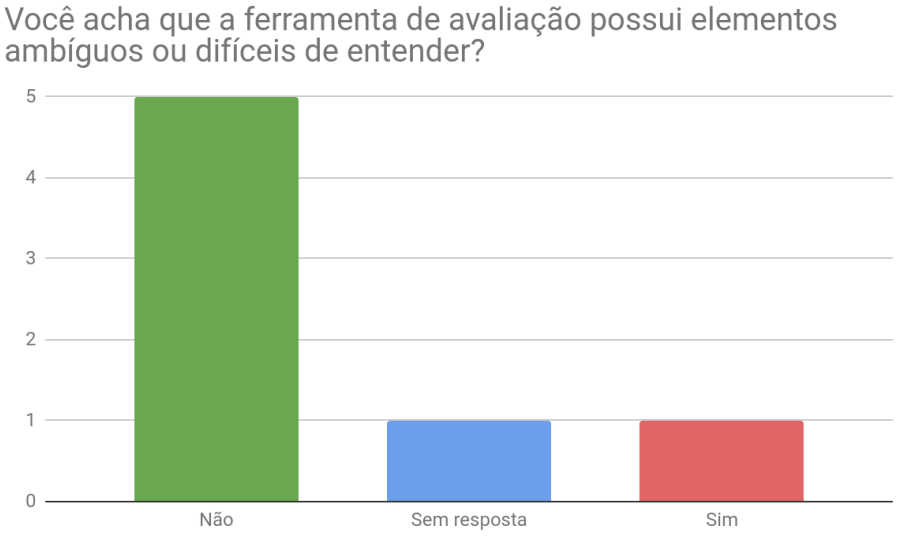


Figura 18 - Resultado da questão de existência de elementos ambíguos na ferramenta

QUAIS OS PONTOS FRACOS E FORTES DO JUPICLASS?

Os resultados das questões abertas sobre quais os pontos fracos e fortes da ferramenta de avaliação e sugestões são apresentados na Tabela 29. A apresentação dos resultados, facilidade de entendimento e velocidade foram citados como pontos fortes. Como fracos, a falta de visualização de quais erros foram cometidos e necessidade de melhorias nas mensagens de erro foram apontadas.

Tabela 29 - Respostas das questões de pontos fracos e fortes e sugestões

Pergunta	Comentários de professores	Comentários de alunos
O que você mais gostou na ferramenta de avaliação?	A possibilidade de poder ajustar os parâmetros do modelo, de ver o modelo funcionando para fazer a classificação, e de poder ter uma avaliação rapidamente ao final do processo. Fácil de entender O que eu mais gostei foi a apresentação dos resultados da avaliação, de forma da figura do ninja e a tabela com os critérios de avaliação. Ficou muito legal todo o processo de criação e avaliação do modelo dentro de um colab. Em geral, está muito bem organizado. Parabéns pelo trabalho!	Um score final compostos pelas avaliações individuais é muito legal, acho que com mais detalhes vai ajudar muito no processo de use-modify-create. É direto ao ponto e quando finalizo e vejo meu desempenho, fica ainda mais claro os processos de treinamento. Facilidade de execução.
O que você menos gostou na ferramenta de avaliação?	A avaliação não deixa claro que erros foram cometidos. Não há o que não tenha gostado Achei que o código necessário para criar os formulários de avaliação e para popular o dicionário pode "poluir" um pouco o código da aplicação de ML no Jupyter. Da ferramenta nada, só não gostei mesmo das minhas primeiras tentativas que não rodou tudo certinho, mas isso, em si, não tem a ver com a ferramenta :)	Alguns conteúdos podem ser melhorados, como mensagens de erro e títulos. Um pouco extensa.
Mais alguma sugestão?	- Na análise da Matriz de Confusão, o uso de acordeons pode não deixar claro para o aluno que é necessário clicar em cada um para selecionar a resposta - Na avaliação da acurácia, seria possível exibir no formulário as categorias na ordem em que aparecem na tabela exibida pelo interp.print_classification_report() ? - Seria interessante ocultar todos os códigos das interfaces gráficas por padrão. Seria interessante também, limpar todas as saídas do Jupyter logo ao iniciar uma nova execução. - Uma sugestão quanto à programação: talvez	

	ficasse mais legal se o jupic já fosse instalado desde o início e, ao invés de ir inserindo os valores em um dicionário, fossem setados atributos de um objeto do jupic. Isso evitaria a necessidade de todo o código para popular o dicionário que acaba “poluindo” o código do modelo. - Os formulários de avaliação poderiam ser gerados dinamicamente pela própria biblioteca jupic, sem a necessidade de inserir o código dos formulários diretamente no Jupyter?	

6.4 DISCUSSÃO

Os resultados da avaliação fornecem uma indicação inicial que a ferramenta jupiclass pode ser útil, funcional, eficiente em desempenho e ter boa usabilidade para a avaliação de treinamentos de modelos de ML para a tarefa de Classificação de Imagens de estudantes do Ensino Médio. A apresentação visual dos resultados com o ninja robô e facilidade de uso da ferramenta foram as principais características positivas levantadas. Como foi observado, uma melhor descrição de como as competências são avaliadas e como a pontuação é gerada faz-se necessária para o entendimento do aluno de como seu treinamento está sendo avaliado. Outro aspecto verificado é que uma avaliação mais completa e com menor atrito na execução por parte dos participantes demanda um ambiente mais controlado, isolado e sem interferências para a execução dos Jupyter Notebooks. Os erros relatados pelos participantes do questionários foram corrigidos.

AMEAÇAS À VALIDADE

Os resultados obtidos nesta avaliação devem ser interpretados com cautela, levando em consideração as ameaças potenciais à sua validade. Devido à falta de medições em um contexto educacional real e/ou com grupo de controle, os resultados são limitados a fornecer apenas uma primeira indicação sobre a qualidade da ferramenta jupiclass. Embora os participantes tenham sido selecionados de forma que seus perfis correspondessem aos usuários em potencial, a falta de mais professores do Ensino Médio e ausência de estudantes desta faixa pode influenciar nos resultados. Sem integrantes do público alvo, possíveis

necessidades de mais explicações podem ter sido relevadas dado o conhecimento prévio dos participantes. Assim, indica-se a necessidade de estudos futuros com maior número de participantes do perfil específicos do público alvo.

Além disso, o tamanho da amostra pode comprometer a generalização dos resultados. O estudo foi baseado em um total de 7 participantes, um tamanho de amostra pequeno e que dificulta qualquer tipo de análise quantitativa. Porém, de acordo com Hakim (1987), pequenas amostras podem ser utilizadas para desenvolver e testar explicações, principalmente nos estágios iniciais do trabalho.

Devido a limitações práticas, os resultados relacionados aos efeitos de aprendizagem foram obtidos a partir de um projeto pré-criado dentro de um contexto artificialmente controlado para fins de avaliação. Esse tipo de avaliação que ocorre fora de um contexto educacional pode não ser suficiente para medir o efeito da ferramenta. Mais estudos de avaliação em contextos educacionais são, portanto, necessários para confirmar os resultados. Visando neutralizar a ameaça de possíveis problemas na definição da medição em si, os questionários foram desenvolvidos decompondo sistematicamente o objetivo da avaliação em itens de questionário que adotam a abordagem GQM (BASILI et al., 1994).

7 CONCLUSÃO

O presente trabalho apresenta o desenvolvimento de um modelo de avaliação de aprendizagem de *Machine Learning* voltado à tarefa de classificação de imagens para o contexto do Ensino Médio. Como parte do TCC foi analisada a fundamentação teórica em relação ao conceito de ensino de ML no Ensino Médio, ML com Jupyter e Python e avaliação de aprendizagem (OE1). Foi levantado também o estado da arte de modelos de avaliação da aprendizagem de ML no contexto do Ensino Médio por meio de um mapeamento sistemático (OE2). Este mapeamento foi publicado como um artigo científico no Workshop em Educação de Computação - WEI 2021 da SBC em conjunto com outros pesquisadores do GQS/INE/UFSC (SALVADOR et al., 2021).

Foi criado um modelo conceitual para avaliar a aprendizagem de ML no Ensino Médio, definindo uma rubrica inédita conforme o estado da arte levantado (OE3). A partir do modelo conceitual, foi criada uma biblioteca automatizando a avaliação com base na rubrica dentro do Jupyter Notebook e gerando um resumo do modelo treinado para fins de documentação (OE4). Também foi realizada uma avaliação preliminar da biblioteca criada com um *expert panel*, gerando resultados iniciais promissores sobre a proposta da ferramenta (OE5).

Desta forma, espera-se realizar uma contribuição importante para o ensino de ML nas escolas Brasileiras, buscando facilitar a avaliação da aprendizagem do aluno e assim contribuir no seu progresso na aprendizagem como um todo.

Como trabalhos futuros pode-se citar a adaptação da ferramenta para tarefas diferentes, como detecção de objetos, a integração da ferramenta com uma interface visual do Jupyter Notebook, e uma melhora validação da rubrica e da fórmula do cálculo de nota aplicado, além da realização de estudos de avaliação mais amplos e em contextos educacionais no Ensino Médio.

REFERÊNCIAS

AI4ALL. **Open Learning brings free and approachable AI education online with the support of Google**. 2018. Disponível em: https://medium.com/ai4allorg/ai4all-open-learning-brings-free-and-accessible-ai-education-online-with-the-support-of-google-org-3a6360c135c9. Acesso em: 18 out. 2020.

AI4K12. **AIK412**. 2020a. Disponível em: https://github.com/touretzkyds/ai4k12/wiki. Acesso em: 17 out. 2020.

AI4K12. **Draft Big Idea 3 - Progression Chart**. 2020b. Disponível em: https://drive.google.com/file/d/1QL6I_I5cdNTVnYBlZ3_Lxur2DgFjmG_d/view. Acesso em 17 out. 2020.

ALA-MUTKA, K. M. **A Survey of Automated Assessment Approaches for Programming Assignments**. Computer Science Education 15(2), pp. 83-102, 2005.

ALPAYDIN, E. **Introduction to Machine Learning**. MIT Press, Massachusetts, USA, 2010.

ALVES, N. C. **CodeMaster: um modelo de avaliação do pensamento computacional na educação básica através da análise de código de linguagem de programação visual**. Dissertação (Programa de Pós-Graduação em Ciência da Computação (PPGCC)) – Universidade Federal de Santa Catarina, 2019a.

ALVES, N. C.; MARQUES, L. S.; GRESSE VON WANGENHEIM, C.; HAUCK, J. C. R. **Approaches to Assess Computational Thinking Competences Based on Code Analysis in K-12 Education: A Systematic Mapping Study**. Informatics in Education 18(1), pp. 17-39, 2019b.

ALVES, N. C.; MARQUES, L. S.; GRESSE VON WANGENHEIM, C.; HAUCK, J. C. R. **A Large-scale Evaluation of a Rubric for the Automatic Assessment of Algorithms and Programming Concepts**. Proc. of the 51st ACM Technical Symposium on Computer Science Education, Portland, OR, USA, 2020.

AMERSHI, S. et al. **Software Engineering for Machine Learning: A Case Study**. Proc. of the IEEE/ACM 41st Int. Conference on Software Engineering: Software Engineering in Practice, Montreal, QC, Canada, 2019.

ANDRADE, E. de L. P. et al. **Katie: saindo do buraco negro e impulsionando as meninas para a computação**. Proc. of the XIV Women in Information Technology 14, Cuiabá, Brasil, pp. 239-243, 2020.

APPS FOR GOOD. **Machine Learning Standard. 2019**. Apps for Good Machine Learning. Disponível em: https://www.appsforgood.org/courses/machine-learning. Acesso em: 19 fev. 2021.

ARAUJO, E. A. C.; ANDRADE, D. F.; BORTOLOTTI, S. L. V. **Teoria de Resposta ao Item**. Revista da Escola de Enfermagem da Universidade de São Paulo, 43, pp. 1000-1008, 2009.

AWS. **Amazon Machine Learning - Splitting the Data into Training and Evaluation Data**. 2020a. Disponível em: https://docs.aws.amazon.com/en_us/machine-learning/latest/dg/splitting-the-data-into-training-and-evaluation-data.html. Acesso em: 2 nov. 2020.

AWS. **What is data labeling for machine learning?** 2020b. Disponível em: https://aws.amazon.com/sagemaker/groundtruth/what-is-data-labeling/. Acesso em: 4 mar. 2021.

BASILI, V. R., CALDIERA, G., ROMBACH, H. D., 1994, **Goal Question Metric Paradigm**. Encyclopedia of Software Engineering, 2 Volume Set, 1994.

BAŞTANLAR, Y.; ÖZUYSAL, M. 2013. **Introduction to Machine Learning**. miRNomics: MicroRNA Biology and Computational Analysis. Methods in Molecular Biology (Methods and Protocols), 1107(1), pp. 105-128, 2013.

BILAL, M., CHAN, P., MEDDINGS, F., & KONSTADOPOULOU, A. **SCORE: An advanced assessment and feedback framework with a universal marking scheme in higher education**. Proc. of the Int. Conference on Education and e-Learning Innovations, Sousse, Tunisia, pp. 1-6, 2012.

BLACK, P.; WILIAN, D. **Assessment and classroom learning**. Assessment in Education: Principles, Policy & Practice 5(1), pp. 7-74, 1998.

BLOOM, B. S.; ENGELHART, M. D.; FURST, E. J.; HILL, W. H.; KRATHWOHL, D. R. **Taxonomy of educational objectives: The classification of educational goals**. Handbook I: Cognitive domain. New York: David McKay Company, 1956.

BRANCH, R. M. **Instructional Design: The ADDIE Approach**. New York: Springer, 2009.

BROOKE, J. **SUS—A Quick and Dirty Usability Scale**. Usability Evaluation in Industry, 189(1), pp. 4-7, 1996.

BRUMMELEN, J. V. et al. **Teaching Tech to Talk: K-12 Conversational Artificial Intelligence Literacy Curriculum and Development Tools**. arXiv:2009.05653, 2020.

CARDOZO, J. **Desenvolvimento de um Curso On-line para o Ensino de Machine Learning no Ensino Médio**. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Universidade Federal de Santa Catarina, Brasil, 2021 (em andamento).

CETIC.BR. **Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nas escolas brasileiras - TIC Educação**. 2014. Disponível em: https://data.cetic.br/cetic/explore?idPesquisa=TIC_EDU&ano=2014. Acesso em: 29 abr. 2021.

CETIC.BR. **Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nas escolas brasileiras - TIC Educação**. 2017. Disponível em: https://data.cetic.br/cetic/explore?idPesquisa=TIC_EDU&ano=2017. Acesso em: 20 abr. 2021.

CHENG, S.; LIN, C.; CHEN, H.,; HEN, J. **Learning and diagnosis of individual and class conceptual perspectives: An intelligent systems approach using clustering techniques**. Computers & Education, 44(3), pp. 257–283, 2005.

CHOLLET, F. **Building powerful image classification models using very little data**. 2016. Disponível em: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html. Acesso em: 3 mar. 2020.

CLICK4IT. 2013. **Assessment**. Disponível em: http://www.click4it.org/index.php/Assessment. Acesso em: 22 nov. 2020.

CME. **Resolução CME Nº02/2011**. Conselho Municipal de Educação de Florianópolis, 2011.

CNE, COMPUTAÇÃO NA ESCOLA. 2020. **CodeMaster – Automatic Assessment and Grading of App Inventor and Snap! Programs**. https://computacaonaescola.ufsc.br/en/codemaster/. Acesso em: 19 jan. 2021.

CODE.ORG. **Lesson 9: AI For Oceans**. 2019. Disponível em: https://curriculum.code.org/hoc/plugged/9/. Acesso em: 19 fev. 2021.

CORDEIRO, A. M.; OLIVEIRA, G. M. de; RENTERÍA, J. M.; GUIMARÃES, C. A. **Revisão sistemática: uma revisão narrativa**. Revista do Colégio Brasileiro de Cirurgiões, 34(6), pp. 428-431, 2007.

CSER. **Teaching Artificial Intelligence in the Secondary Classroom**. Disponível em: https://csermoocs.appspot.com/ai_secondary/. Acesso em: 23 mar. 2021.

CSTA. **CSTA K-12 Computer Science Standards**. Association for Computing Machinery, New York, NY, USA, 2011.

CSTA. **CSTA K-12 Computer Science Framework**. 2016. Disponível em: https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf. Acesso em: 17 out. 2020.

CSTA. **CSTA K-12 Computer Science Standards**. 2017. Disponível em: http://www.csteachers.org/standards. Acesso em: 17 out. 2020.

DAGIENE, V. **Teaching Information Technology and Elements of Informatics in Lower Secondary Schools: Curricula, Didactic Provision and Implementation**. Proc. of the 3rd Int. Conference on Informatics in Secondary Schools - Evolution and Perspectives, Torun, Poland, pp. 293–304, 2008.

DARGAN, S.; KUMAR, M.; AYYAGARI, M. R.; KUMAR, G. **A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning**. Archives of Computational Methods in Engineering, 27(4), pp. 1071-1092, 2020.

DAVIS, F. D. **Perceived usefulness, perceived ease of use, and user acceptance of information technology**. MIS Quarterly, 13(3), pp. 319–340, 1989.

DEEPAI. **What is Machine Learning?** 2019. Disponível em: https://deepai.org/machine-learning-glossary-and-terms/machine-learning. Acesso em: 2 nov. 2020.

DENNING, Peter J.; TEDRE, Matti. **Computational Thinking**. Massachusetts: MIT Press, 2019.

DOCKER. **Docker**. Disponível em: https://www.docker.com/. Acesso em: 22 ago. 2021.

ELEARNING INDUSTRY. **5 Guidelines For Developing Good Online Assessments**. 2018. Disponível em:

https://elearningindustry.com/developing-good-online-assessments-guidelines. Acesso em: 21 jan. 2021.

ELEMENTS OF AI. **Machine Learning**. 2019. Disponível em: https://course.elementsofai.com/4. Acesso em: 21 fev. 2021.

ESTEVES, J.; GARATE, G.; GRAÑA, M. **Gentle Introduction to Artificial Intelligence for High-School Students Using Scratch**. IEEE Access, 7(1), pp. 179027-179036, 2019.

EVANGELISTA, I.; BLESIO, G.; BENATTI, E. **Why Are We Not Teaching Machine Learning at High School? A Proposal.** Proc. of the World Engineering Education Forum, Albuquerque, NM, USA, 2018.

EXPLORING COMPUTER SCIENCE. 2019. **Artificial Intelligence - Alternate Curriculum Unit**. Disponível em: http://www.exploringcs.org/for-teachers-districts/artificial-intelligence. Acesso em: 8 out. 2020.

FAYYAD, U. PIATESTSKY-SHAPIRO, G.; SMYTH., P. **The KDD process for extracting useful knowledge from volumes of data**. Communications of the ACM, 39(11), pp. 27-34, 1996.

FLASK. **Flask Documentation**. Disponível em: https://flask.palletsprojects.com/en/2.0.x/. Acesso em: 22 ago. 2021.

FORBES. 2019. **AI Goes to High School**. Disponível em: https://www.forbes.com/sites/insights-intelai/2019/05/22/ai-goes-to-high-school. Acesso em: 25 out. 2020.

GAL-ZER, J.; BEERI, C.; HAREL, D.; YEHUDAI, A.. **A High School Program in Computer Science.** Computer, 28(10), pp. 73–80, 1995.

GALAN, D.; HERADIO, R.; VARGAS, H.; ABAD, I.; CERRADA, J. A. **Automated Assessment of Computer Programming Practices: The 8-Years UNED Experience**. IEEE Access, 7(1), pp. 130113-130119, 2019.

GOFORTH, Chelsea. **Using and Interpreting Cronbach's Alpha**. University of Virginia, 2015.

GOGTAY, N. J.; THATTE, U. M. **Principles of Correlation Analysis**. Journal of The Association of Physicians of India, 65(3), pp. 78-81, 2017.

GOLAFSHANI, Nahid. **Understanding Reliability and Validity in Qualitative Research**. The Qualitative Report, 8(4), pp. 597-607, 2003.

GOOGLE. **Google Teachable Machine**. 2020a. Disponível em: https://teachablemachine.withgoogle.com/. Acesso em: 7 out. 2020.

GOOGLE. **Descending into ML: Training and Loss.** 2020b. Disponível em: https://developers.google.com/machine-learning/crash-course/descending-into-ml/training-and-loss. Acesso em: 5 nov. 2020.

GOOGLE. **Reducing Loss: Stochastic Gradient Descent**. 2020c. Disponível em: https://developers.google.com/machine-learning/crash-course/reducing-loss/stochastic-gradient-descent. Acesso em: 7 nov. 2020.

GOOGLE. **O que é o Colaboratory?** 2020d. Disponível em: https://colab.research.google.com/notebooks/intro.ipynb. Acesso em: 8 nov. 2020.

GOOGLE CLOUD. **Machine learning workflow**. 2018. Disponível em: https://cloud.google.com/ai-platform/docs/ml-solutions-overview. Acesso em: 4 mar. 2020.

GRESSE VON WANGENHEIM, C; MARQUES, L. S.; HAUCK, J. C. R. **Machine Learning for All – Introducing Machine Learning in K-12**. SocArXiv, 2020.

GRESSE VON WANGENHEIM, C. **Overview on a human-centric interactive ML process for teaching ML in K-12**. Working Paper WP_GQS_01_2021_v10, GQS/INCoD/UFSC, 2021.

GUO, Y.; SHI, H.; KUMAR, A.; GRAUMAN, K.; ROSING, T.; FERIS, R. **SpotTune: Transfer Learning Through Adaptive Fine-Tuning**. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, California, USA, 2019.

GUPTA, T. **Deep Learning: Feedforward Neural Network**. 2017. Disponível em: https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7. Acesso em: 7 nov. 2020.

GURESEN, E.; KAYAKUTLU, G. **Definition of artificial neural networks with comparison to other networks**. Procedia Computer Science, 3, 2011.

HAKIM, C. **Research Design: Strategies and Choices in the Design of Social Research**. Contemporary Social Research Series 13. Allen and Unwin, London, UK,

1987

HATTIE, J; TIMPERLEY, H. **The power of feedback**. Review of Educational Research, 77(1), pp. 81-112, 2007.

HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep Residual Learning for Image Recognition**. arXiv:1512.03385, 2015.

HILL, H.; ROWAN, D.; BALL, D. **Effects of teachers' mathematical knowledge for teaching on student achievement**. American Educational Research Journal, 42(2), pp. 371-406, 2005.

HO, J. W. K.; SCADDING, M. **Classroom Activities for Teaching Artificial Intelligence to Primary School Students**. Proc. of the Int. Conference on Computational Thinking, Hong Kong, China, 2019.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D. **MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications**. arXiv:1704.04861v1, 2017.

HOWARD, J.; GUGGER, S. **Fastai: A Layered API for Deep Learning**. Information, 11(2), 2020.

HUBWIESER, P. et al. **A Global Snapshot of Computer Science Education in K-12 Schools**. Proc. of the ITiCSE on Working Group Reports, Vilnius, Lithuania, 2015.

INEP. 2018. **Notas estatísticas - Censo Escolar 2018**. Disponível em: https://download.inep.gov.br/educacao_basica/censo_escolar/notas_estatisticas/2018/notas_estatisticas_censo_escolar_2018.pdf. Acesso em: 27 mar. 2021.

IPYWIDGETS. **ipywidgets**. Disponível em: https://ipywidgets.readthedocs.io/en/stable/. Acesso em: 20 abr. 2021.

ISO 9241. **ISO 9241-11:1998(en) - Guidance on usability**. International Organization for Standardization, 1998.

ISO/IEC 25010. **ISO/IEC 25010:2011**, **Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE)**. International Organization for Standardization, 2011.

ISTE. **Bold New Program Helps Teachers and Students Explore the Power of AI**.

2018. Disponível em: https://www.iste.org/explore/Press-Releases/Bold-New-Program-Helps-Teachers-and-Students-Explore-the-Power-of-AI. Acesso em: 18 out. 2020.

JING, M. 2018. **China looks to school kids to win the global AI race**. Disponível em: https://www.scmp.com/tech/china-tech/article/2144396/china-looks-school-kids-win-global-ai-race. Acesso em: 17 out. 2020.

KAHN, K. M.; LU, Y.; ZHANG, J.; WINTERS, N.; GAO, M. **Deep learning programming by all.** Proc. of the Conference on Constructionism, Dublin, Ireland, 2020.

KANDLHOFER, M.; STEINBAUER G.; HIRSCHMUGL-GAISCH S.; HUBER, P. **Artificial Intelligence and Computer Science in Education: From Kindergarten to University**. Proc. of the IEEE Frontiers in Education Conference, Erie, PA, USA, 2016.

KNEKTA, E.; RUNYONG, C.; EDDY, S. **One Size Doesn't Fit All: Using Factor Analysis to Gather Validity Evidence When Using Surveys in Your Research**. CBE: Life Sciences Education, 2019.

KONG, S.C.; HOPPE, H.U; HSU, T.C.; HUANG, R.H.; KUO, B.C.; LI, K.Y.; LOOI, C.K.; MILRAD, M.; SHIH, J.L.; SIN, K.FL; SONG, K.S.; SPECHT, M.; SULLIVAN, F.; VAHRENHOLD, J. **Experiences from Teaching Actionable Machine Learning at the University Level through a Small Practicum Approach**. Proc. of Int. Conference on Computational Thinking Education, Hong Kong, 2020.

KRETLOW, A., & BARTHOLOMEW, C. **Using coaching to improve the fidelity of evidence-based practices: A review of studies**. Teacher Education and Special Education, 33(4), pp. 279–299, 2010.

LALOR, J. P; WU, H. YU, H. **Building an Evaluation Scale using Item Response Theory.** Proc. of the Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, pp. 648-657, 2016.

LARMAN, C.; BASILI, V. R. **Iterative and Incremental Development: A Brief History**. Computer, 36(6), pp. 47-56, 2003.

LEE, I. et al. **Developing Middle School Students' AI Literacy**. Proc. of the ACM Special Interest Group on Computer Science Education, pp. 191-197, 2021.

LIMESURVEY. **LimeSurvey**. 2021. Disponível em: https://www.limesurvey.org/pt/. Acesso em 16 set. 2021.

LOBATO, A. S. et al. **Uma rubrica para avaliação de cursos de programação centrada em avaliação automática**. Proc. of the Simpósio Brasileiro de Informática na Educação - Workshop de Ambientes de apoio à Aprendizagem de Algoritmos e Programação, São Paulo, Brasil, 2007.

LONG, D.; MAGERKO, B. **What is AI Literacy? Competencies and Design Considerations**. Proc. of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, pp. 1-16, 2020.

LWAKATARE, L.E.; RAJ, A.; BOSCH, J.; OLSSON, H. H.; CRNKOVIC, I. **A Taxonomy of Software Engineering Challenges for Machine Learning Systems: An Empirical Investigation**. Proc. of the Agile Processes in Software Engineering and Extreme Programming, Montréal, QC, Canada, 2019.

LYE, S. Y.; KOH, J. H. L. **Review on teaching and learning of computational thinking through programming: What is next for K-12?**. Computers in Human Behavior, 41(1), pp. 51–61, 2014.

LYTLE, N. et al. **Use, Modify, Create: Comparing Computational Thinking Lesson Progressions for STEM Classes.** Proc. of the Conference on Innovation and Technology in Computer Science Education, ACM, pp. 395–401, 2019.

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), pp. 283–321, 2020.

MARTINS, L. DA C. G. F.; GUISSO, L. F. **Avaliação: um desafio no processo de ensino-aprendizagem na educação - revisão de literatura**. Revista Eletrônica Acervo Saúde, 24, e379. 2019.

MARTINS, O. P. H. R. **Desenvolvimento de um Modelo para Avaliação da Estética Visual de Interfaces de Usuários de Aplicativos Usando Deep Learning**. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade Federal de Santa Catarina, Brasil, 2019.

MATPLOTLIB. **Matplotlib**. 2020. Disponível em: https://matplotlib.org/. Acesso em: 8 nov. 2020.

MCGOVERN, A.; TIDWELL, Z.; RUSHING, D. **Teaching Introductory Artificial**

Intelligence through Java-Based Games. Proc. of the 2nd Symposium on Educational Advances in Artificial Intelligence, San Francisco, CA, USA, 2011.

MINISTÉRIO DA EDUCAÇÃO. 2017. **Base Nacional Comum Curricular**. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 25 out. 2020.

MINISTÉRIO DA EDUCAÇÃO. 2018. **O jovem no Ensino Médio**. Disponível em: http://portal.mec.gov.br/dia-a-dia-do-seu-filho/o-jovem-no-ensino-medio. Acesso em: 27 mar. 2021.

MISLEVY, R. J.; ALMOND, R. G.; LUKAS, J. F. **A Brief Introduction to Evidence-centered Design**. Educational Testing Service, Research & Development Division, Princeton, NJ, USA. 2003.

MIT APP INVENTOR. **Introduction to Machine Learning: Image Classification**. 2019. Disponível em: https://appinventor.mit.edu/explore/resources/ai/image-classification-look-extension. Acesso em: 19 fev. 2021.

MITCHELL, M. et al. **Model Cards for Model Reporting**. arXiv:1810.03993, 2019.

MONTALTI, M. **Building a culture of feedback: Teacher to Student – Student to Student – Student to Teacher**. e-Teaching 27, Australian Council for Educational Leaders, Australia, 2016.

MORAES, J. B.; RODRIGUES, M. S. **Competências profissionais: um estudo comparativo entre os cursos superiores de tecnologia em processos gerenciais do Instituto Federal de São Paulo**. Latin American Journal of Business Management, 10(2), 2019.

MORENO, R. **Decreasing Cognitive Load for Novice Students: Effects of Explanatory versus Corrective Feedback in Discovery-Based Multimedia**. Instructional Science, 32(1), pp. 99–113, 2004.

MORENO-GER, P.; BURGOS, D.; MARTÍNEZ-ORTIZ, I.; SIERRA, J. L.; FERNÁNDEZ-MANJÓN, B. **Educational game design for online education**. Computers in Human Behavior, 24 (6), pp. 2530-2540, 2008.

NAH, F. F. H.; ZENG, Q.; TELAPROLU, V. R.; AYYAPPA, A. P.; ESCHENBRENNER, B. **Gamification of Education: A Review of Literature**. Proc. of the Int. Conference

on Human-Computer Interaction in Business, Heraklion, Crete, Greece, pp. 401-409, 2014.

NARCISS, S.; HUTH, K. **How to design informative tutoring feedback for multi-media learning**. Instructional Design for Multimedia learning, New York: Waxmann, pp. 181-195, 2004.

NBVIEWER. **nbviewer**. 2020. Disponível em: https://nbviewer.jupyter.org/. Acesso em: 8 nov. 2020.

NOVAKOVIĆ, J. D. et al. **Evaluation of Classification Models in Machine Learning.** Theory and Applications of Mathematics & Computer Science, 7(1), Belgrade, Serbia, pp. 39-46, 2017.

NVIDIA. **NVIDIA Techsplorer**. 2020. Disponível em: https://www.nvidia.com/en-us/foundation/programs/techsplorer-stem-program/. Acesso em: 14 nov. 2020.

NUMPY. **NumPy**. 2020. Disponível em: https://numpy.org/. Acesso em 8 nov. 2020.

OLIVEIRA, A. P. S. B. de; PONTES, J. N. de A.; MARQUES, M. A. **O Uso da Taxionomia de Bloom no Contexto da Avaliação por Competência**. Pleiade, 10(20), 2016.

PAPATHOMA, T. **Investigating Different Types of Assessment in Massive Open Online Courses**. MRes thesis - The Open University, UK, 2015.

PERKEL, J. M. **Why Jupyter is data scientists' computational notebook of choice**. Nature, 563(7729), pp. 145-146, 2018.

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. **Systematic Mapping Studies in Software Engineering**. Proc. of the 12th Int. Conference on Evaluation and Assessment in Software Engineering, Bari, Italy, pp. 68-77, 2008.

PIASECKI, J.; WALIGORA, M.; DRANSEIKA, V. **Google Search as an Additional Source in Systematic Reviews**. Science and Engineering Ethics, 24(2), pp. 809-910, 2018.

POETRY. **Poetry - Python dependency management and packaging made easy**. 2021. Disponível em: https://python-poetry.org/. Acesso em: 24 ago. 2021.

POWERS, D. M. W. **Evaluation: from precision, recall and F-measure to ROC,**

informedness, markedness and correlation. arXiv:2010.16061, 2020.

PRADO, J. C.; MARZAL, M. Á.. **Incorporating Data Literacy into Information Literacy Programs: Core Competencies and Contents**. Libri, 63 (2), pp. 123-134, 2013.

PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. Porto Alegre: AMGH, 2016.

PUTZ, L. M.; HOFBAUER, F.; TREIBLMAIER, H. **Can gamification help to improve education? Findings from a longitudinal study.** Computers in Human Behavior, 110(1), p. 106392, 2020.

PYPI. **PyPI - The Python Package Index**. 2021. Disponível em: https://pypi.org/. Acesso em: 16 set. 2021.

RAGUPATHI, Kiruthika. **Designing Effective Online Assessments**. Resource Guide - National University of Singapore, Singapore. 2020.

RAMAPRASAD, A. **On the Definition of Feedback**. Systems Research and Behavioral Science 28(1), pp. 4-13, 1983.

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, 35(5-6), pp. 413-451, 2020.

RANDLES, B. M.; PASQUETTO, I. V.; GOLSHAN, M. S.; BORGMAN, C. L. **Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study**. Proc. of the ACM/IEEE Joint Conference on Digital Libraries, Ontario, Canada, 2017.

READYAI. **AI + ME**. 2019. Disponível em: https://edu.readyai.org/pt-pt/courses/ai-me-1/. Acesso em: 19 fev. 2021.

ROBERTS, Paula; PRIEST, Helena. **Reliability and validity in research**. Nursing Standard, 20(44), 2006.

RODRÍGUEZ-GARCÍA, J. D.; MORENO-LEÓN, J.; ROMÁN-GONZÁLEZ, M.; ROBLES, G. 2020. **LearningML: A Tool to Foster Computational Thinking Skills Through Practical Artificial Intelligence Projects**. Distance Education Journal, 20(63), 2020.

RODRÍGUEZ-GARCÍA, J. D.; MORENO-LEÓN, J.; ROMÁN-GONZÁLEZ, M.; ROBLES, G. 2021. **Evaluation of an Online Intervention to Teach Artificial**

Intelligence With LearningML to 10-16-Year-Old Students. Proc. of the ACM Special Interest Group on Computer Science Education Technical Symposium, Toronto, Canada, 2021.

RUSSAKOVSKY, O. et al. **ImageNet Large Scale Visual Recognition Challenge**. Int. Journal of Computer Vision, 115(3), pp. 211–252, 2015.

SAKULKUEAKULSUK, B. et al. **Kids making AI: Integrating Machine Learning, Gamification, and Social Context in STEM Education**. Proc. of the IEEE Int. Conference on Teaching, Assessment, and Learning for Engineering, Wollongong, NSW, Australia, pp. 1005-1010, 2018.

SALVADOR, G.; GRESSE VON WANGENHEIM, C.; RAUBER, M. F.; GARCIA, A. B.; BORGATTO, A. F. **Avaliação de Aprendizagem de Machine Learning na Educação Básica: Um Mapeamento da Literatura**. CSBC 2021 - WEI, 2021.

SAMUEL, A. L. **Some Studies in Machine Learning Using the Game of Checkers**. IBM Journal of Research and Development, 44(1-2), pp. 210–229, 2000.

SBC. **Diretrizes para ensino de Computação na Educação Básica**. 2018a. Disponível em: https://www.sbc.org.br/documentos-da-sbc/send/203-educacao-basica/1220-bncc-em-itinerario-informativo-computacao-2. Acesso em: 23 dez. 2020.

SBC. **Nota Técnica da Sociedade Brasileira de Computação sobre a BNCC-EF e a BNCC-EM**. 2018b. Disponível em: https://www.sbc.org.br/institucional-3/cartas-abertas/send/93-cartas-abertas/1197-nota-tecnica-sobre-a-bncc-ensino-medio-e-fundamental. Acesso em: 25 out. 2020.

SCIKIT-LEARN. **Scikit-learn**. 2020. Disponível em: https://scikit-learn.org/. Acesso em: 18 out. 2020.

SEWELL, J.; FRITH, K. H.; COLVIN, M. M. **Online assessment strategies: A primer**. Journal of Online Learning and Teaching, 6(1), pp. 297-305, 2010.

SEWELL, J.; THEDE, L. Q. **Informatics and Nursing**. Philadelphia: Wolters Kluwer, 2009.

SHARIQ, M.; PERERA, M. **Virtual classroom simulation using agent technology**. Proc. of the Int. Conference on Educational and Information Technology, Chongqing, China, pp. 197-201, 2010.

SODHI, P.; AWASTHI, N.; SHARMA, V. **Introduction to Machine Learning and Its Basic Application in Python**. Proc. of 10th Int. Conference on Digital Strategies for Organizational Success, Gwalior, MP, India, 2019.

SOLECKI, I.; PORTO, J. A.; ALVES, N. d. C., GRESSE VON WANGENHEIM, C., HAUCK, J. C. R., BORGATTO, A. F. **Automated Assessment of the Visual Design of Android Apps Developed with App Inventor.** Proc. of the 51st ACM Technical Symposium on Computer Science Education, Portland, OR, USA, pp. 51–57, 2020.

STEGEMAN, M.; BARENDSEN, E.; SMETSERS, S. **Designing a rubric for feedback on code quality in programming courses**. Proc. of the 16th Koli Calling Int. Conference on Computing Education Research, Koli, Finland, pp. 160-164, 2016.

STEVENS, D. D.; LEVI, A. J. **Introductions to rubrics: an assessment tool to save grading time, convey effective feedback and promote student learning**. Virginia: Stylus Pub., 2005.

SUSKIE, L. **Assessing student learning: A common sense guide**. San Francisco: Jossey-Bass, 2009.

SYSLO, M. M. **Outreach to Prospective Informatics Students**. Informatics in Schools. Proc. of the 5th International Conference on Informatics in Schools: Situation, Evolution and Perspectives, Bratislava, Slovakia, pp. 56–70, 2011.

TANG, D.; UTSUMI, Y.; LAO, N. **PIC: A Personal Image Classification Webtool for High School Students**. Proc. of the Int. Joint Conferences on Artificial Intelligence EduAI Workshop, Macao, China, 2019.

TANG, D. **Empowering novices to understand and use machine learning with personalized image classification models, intuitive analysis tools, and MIT App Inventor**. Thesis: M. Eng., Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2019.

TAVAKOL, M.; DENNICK, R. **Making sense of Cronbach's alpha**. Int. Journal of Medical Education, 2(1), pp. 53-55, 2011.

TECHGIRLZ. **Artificial Intelligence: How Computers Learn**. 2020. Disponível em: https://www.techgirlz.org/topic/artificial-intelligence-computers-learn/. Acesso em: 17 out. 2020.

TECHNOVATION FAMILIES. **AI Family Challenge**. 2020. Disponível em: https://www.curiositymachine.org/about/. Acesso em: 8 out. 2020.

TENSORFLOW. **Tensorflow Playground**. 2020. Disponível em: https://playground.tensorflow.org/. Acesso em: 18 out. 2020.

TISSENBAUM, M.; SHELDON, J.; ABELSON, H. **From computational thinking to computational action**. Communications of the ACM, 62(3), 2019.

TORREY, L. **Teaching Problem-Solving in Algorithms and AI**. Proc. of the 3rd Symposium on Educational Advances in Artificial Intelligence, Toronto, Ontario, Canada, 2012.

TOURETZKY, D. S. et al. **Envisioning AI for K-12: What Should Every Child Know about AI?** Proc. of the 33rd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, Honolulu, HI, USA, 2019a.

TOURETZKY, D. S.; GARDNER-MCCUNE, C.; MARTIN, F.; SEEHORN, D. **K-12 Guidelines for Artificial Intelligence: What Students Should Know**. Proc. of the Int. Society for Technology in Education Conference, Philadelphia, PA, USA, 2019b.

TYLER, R. W. **Basic Principles of Curriculum and Instruction**. Chicago: University of Chicago Press, 1949.

WANG, H.; MA, C.; ZHOU, L. **A Brief Review of Machine Learning and Its Application**. Proc. of the Int. Conference on Information Engineering and Computer Science, Wuhan, China, pp. 1-4, 2009.

WANG, Y., YOUNG, S., WEI, C. **The impact of e-learning on school teachers in recurrent education**. Proc. of the Int. Conference on Electrical and Control Engineering, Yichang, China, pp. 6920-6924, 2011.

WARDEN, P. **How many images do you need to train a neural network?** 2017. Disponível em: https://petewarden.com/2017/12/14/how-many-images-do-you-need-to-train-a-neural-network/. Acesso em: 2 mar. 2021.

WIRTH, R.; HIPP, J. **CRISP-DM: Towards a standard process model for data mining**. Proc. of the Int. Conference on Practical Applications of Knowledge Discovery and Data mining, Manchester, UK, 2000.

WONG, G. K. W.; MA, X.; DILLENBOURG, P.; HUAN, J. **Broadening artificial intelligence education in K-12: where to start?** ACM Inroads, 111(1), pp. 20-29, 2020.

ZIEKY, M. J. **An introduction to the use of evidence-centered design in test development**. Psicología Educativa, 20(2), pp. 79-87, 2014

APÊNDICE 1 - EXEMPLO DE JSON GERADO


```
{
    "model_categories": [
        "Aroeira",
        "Jeriva",
        "Pitangueira",
        "Embauba",
        "Mulungu",
        "Capororoca"
    ],
    "model_correctly_labeled_images": 100,
    "dataset_categories_images": [
        {
            "Aroeira": 20
        },
        {
            "Capororoca": 20
        },
        {
            "Embauba": 20
        },
        {
            "Jeriva": 20
        },
        {
            "Mulungu": 20
        },
        {
            "Pitangueira": 20
        }
    ],
    "dataset_total_images": 100,
    "tl_models": [
        "resnet50",
        "resnet32"
    ],
    "tl_epochs": [
        1,
        2
    ],
    "tl_learning_rates": [
        0.0,
        0.0
    ],
    "tl_trained": true,
    "tl_accuracy_categories": [
        {
            "Aroeira": 0.9
```

```
        },
        {
            "Capororoca": 0.9
        },
        {
            "Embauba": 0.9
        },
        {
            "Jeriva": 0.9
        },
        {
            "Mulungu": 0.9
        },
        {
            "Pitangueira": 0.9
        }
    ],
    "tl_accuracy_analysis": true,
    "tl_accuracy_analysis_categories": [
        "Aroeira",
        "Capororoca",
        "Embauba",
        "Jeriva",
        "Mulungu",
        "Pitangueira"
    ],
    "tl_accuracy_interpretation": false,
    "tl_confusion_matrix_mislabeled_real": [
        {
            "Pitangueira": [
                "Capororoca",
                "Aroeira",
                "Embauba",
                "Mulungu"
            ]
        },
        {
            "Jeriva": []
        },
        {
            "Capororoca": [
                "Jeriva",
                "Aroeira"
            ]
        },
        {
            "Mulungu": []
        },
```

```
        {
            "Embauba": []
        },
        {
            "Aroeira": [
                "Pitangueira",
                "Embauba",
                "Jeriva",
                "Mulungu"
            ]
        }
    ],
    "tl_confusion_matrix_mislabeled": [
        {
            "Pitangueira": [
                "Aroeira",
                "Capororoca",
                "Embauba",
                "Mulungu"
            ]
        },
        {
            "Jeriva": []
        },
        {
            "Capororoca": [
                "Aroeira",
                "Jeriva"
            ]
        },
        {
            "Mulungu": []
        },
        {
            "Embauba": []
        },
        {
            "Aroeira": [
                "Embauba",
                "Jeriva",
                "Mulungu",
                "Pitangueira"
            ]
        }
    ],
    "tl_confusion_matrix_interpretation": false,
    "ft_unfreezed": true,
    "ft_learning_rate_found": true,
```

```
    "ft_trained": true,
    "ft_accuracy_categories": [
        {
            "Aroeira": 0.1
        },
        {
            "Capororoca": 0.2
        },
        {
            "Embauba": 0.3
        },
        {
            "Jeriva": 0.4
        },
        {
            "Mulungu": 0.5
        },
        {
            "Pitangueira": 0.6
        }
    ],
    "ft_accuracy_analysis": false,
    "ft_accuracy_analysis_categories": [
        "Aroeira",
        "Capororoca",
        "Embauba",
        "Jeriva",
        "Mulungu",
        "Pitangueira"
    ],
    "ft_accuracy_interpretation": false,
    "ft_confusion_matrix_mislabeled_real": [
        {
            "Pitangueira": [
                "Capororoca",
                "Aroeira",
                "Embauba",
                "Mulungu"
            ]
        },
        {
            "Jeriva": []
        },
        {
            "Capororoca": [
                "Jeriva",
                "Aroeira"
            ]
```

```
        },
        {
            "Mulungu": []
        },
        {
            "Embauba": []
        },
        {
            "Aroeira": [
                "Pitangueira",
                "Embauba",
                "Jeriva",
                "Mulungu"
            ]
        }
    ],
    "ft_confusion_matrix_mislabeled": [
        {
            "Pitangueira": [
                "Aroeira",
                "Capororoca",
                "Embauba",
                "Mulungu"
            ]
        },
        {
            "Jeriva": []
        },
        {
            "Capororoca": [
                "Aroeira",
                "Jeriva"
            ]
        },
        {
            "Mulungu": []
        },
        {
            "Embauba": []
        },
        {
            "Aroeira": [
                "Embauba",
                "Jeriva",
                "Mulungu",
                "Pitangueira"
            ]
        }
```

```
    ],
    "ft_confusion_matrix_interpretation": false,
    "performance_tuning": 1,
    "performance_tuning_text": "Acurácia",
    "real_objects": [
        "Aroeira",
        "Aroeira",
        "Aroeira",
        "Aroeira",
        "Aroeira"
    ],
    "predicted_objects": [
        "Aroeira",
        "Aroeira",
        "Aroeira",
        "Aroeira",
        "Aroeira"
    ],
    "predicted_success_times": 5,
    "predicted_success_interpretation": true
}
```

APÊNDICE 2 - QUESTIONÁRIO DA AVALIAÇÃO

Avaliação da ferramenta de avaliação de modelos de ML para Classificação de Imagens com Jupyter

Olá,

Gostaríamos de convidar você para participar da avaliação da ferramenta *jupiclass*, que permite automaticamente avaliar o desenvolvimento de modelos de *Machine Learning* para classificação de imagens criados em *Jupyter Notebooks* no contexto do ensino de ML no ensino médio. A ferramenta de avaliação foi desenvolvida no contexto do TCC de Gustavo Salvador sob a orientação da Profa. Christiane Gresse von Wangenheim e Prof. Jean Hauck, realizado na iniciativa de Computação na Escola no INCoD/INE/UFSC.
Na avaliação estaremos solicitando a você que execute o desenvolvimento de um modelo de ML (já programado no *Jupyter Notebook*) para a classificação de 6 espécies de árvores nativas, avalie seu desempenho utilizando a ferramenta de avaliação automatizada *jupiclass* e ao final responda um questionário sobre a utilidade e usabilidade da ferramenta.
No total, não deve levar mais do que 30 minutos. Todos os seus dados serão tratados de forma sigilosa usados somente para fins de pesquisa.
Desde já, muito obrigado pela ajuda! O seu *feedback* é muito importante para nossa pesquisa. Ao final estaremos disponibilizando a extensão de forma aberta e gratuita no site da iniciativa Computação na Escola para qualquer interessado. Assim, solicitamos que esta versão atual (protótipo) ainda não seja compartilhada com outras pessoas.

Att,
Gustavo, Jean e Christiane

Você concorda em participar da avaliação da ferramenta?

Sim | Não | Sem resposta

Por favor, preencha as informações abaixo:

Qual o seu nome completo?

Você atua como:

Escolha a(s) que mais se adeque(m)

- [ ] Professor(a) no ensino médio
- [ ] Professor(a) na graduação
- [ ] Professor(a) na pós-graduação
- [ ] Aluno(a) na graduação
- [ ] Aluno(a) na pós-graduação

Você já desenvolveu quantos modelos de ML para classificação de imagens com Jupyter Notebook?

Escolha uma das seguintes respostas:

- ○ 0
- ○ 1
- ○ 2
- ○ 3-5
- ○ Mais do que 5
- ● Sem resposta

Aos professores(as) ensinando ML nas suas disciplinas: como você atualmente avalia os resultados de trabalhos práticos voltados a criação de modelos para classificação de imagens?

Vamos começar a avaliação!

1. Faça login na conta Google abaixo:
E-mail: jupic.assessment.gqs@gmail.com
Senha: 95wk"XMyZ;}.H)8E

2. Abra o Jupyter Notebook via Google Colab.

3. Preparamos o Notebook para representar a execução por um estudante de ensino médio iniciante em *Machine Learning,* com conjunto de dados já coletado e disponibilizado no Google Drive da mesma conta acima.
Para simular o desenvolvimento do modelo para a classificação de 6 espécies de árvores com os comandos já programados no Google Colab, basta executar as células de código das seções do notebook e responder às células de perguntas (selelção de modelo, quantidade de épocas, questões de interpretação, etc.), da seção *ANÁLISE DE REQUISITOS* até a *EXPORTAÇÃO*.

4. Ao final do desenvolvimento, execute a avaliação automatizada na seção *AVALIAÇÃO DA APRENDIZAGEM COM BASE NO DESEMPENHO*. Veja o resultado do desempenho no modelo e depois passe para a próxima página.

Você conseguiu efetuar o treinamento do modelo no Google Colab?

✔ Sim | ⊘ Não | ○ Sem resposta

	Concordo totalmente	Concordo	Não concordo nem discordo	Discordo	Discordo totalmente	Sem resposta
Acho a ferramenta de avaliação útil na educação de computação no ensino médio.	○	○	○	○	○	●
Acho a ferramenta de avaliação útil no ensino de computação para iniciantes em cursos de graduação.	○	○	○	○	○	●
Acho a ferramenta de avaliação útil na educação de computação para iniciantes em cursos de pós-graduação.	○	○	○	○	○	●
Acho que em sua forma atual a ferramenta de avaliação pode ser aplicada de forma prática em minhas aulas.	○	○	○	○	○	●

Você acha que existem aspectos/critérios para avaliar a construção de modelos de ML para Classificação de Imagens no contexto de ensino de ML no ensino médio que não são suportados pela ferramenta?

Escolha uma das seguintes respostas:

- ○ Sim (se sim, qual?)
- ○ Não
- ● Sem resposta

Por favor, coloque aqui o seu comentário:

Você acha que existem aspectos relevantes no que diz respeito ao processo de avaliação da construção de modelos de ML para Classificação de Imagens no contexto de ensino de ML no ensino médio que não são apoiados pela ferramenta de avaliação?

Escolha uma das seguintes respostas:

- ○ Sim (se sim, qual?)
- ○ Não
- ● Sem resposta

Por favor, coloque aqui o seu comentário:

Você acha que as informações de feedback fornecidas são suficientes?

Escolha uma das seguintes respostas:

- ○ Sim
- ○ Não (se não, o que está faltando?)
- ◉ Sem resposta

Por favor, coloque aqui o seu comentário:

Você notou algum erro em relação à funcionalidade da ferramenta de avaliação?

Escolha uma das seguintes respostas:

- ○ Sim (se sim, qual?)
- ○ Não
- ◉ Sem resposta

Por favor, coloque aqui o seu comentário:

Você achou os resultados da avaliação corretos?

Escolha uma das seguintes respostas:

- ○ Sim
- ○ Não (se não, explique)
- ◉ Sem resposta

Por favor, coloque aqui o seu comentário:

O desempenho em termos de tempo de processamento da ferramenta de avaliação é aceitável (não leva muito tempo)?

Sim	Não	Sem resposta
✔	⊘	○

Você conseguiu concluir uma avaliação usando a ferramenta de avaliação?

Sim	Não	Sem resposta
✔	⊘	○

Você considera que o tempo que leva para interagir com a ferramenta de avaliação para obter feedback é adequado (não leva muito tempo)?

Sim	Não	Sem resposta
✔	⊘	○

Comparando o esforço/tempo que você levou para avaliar o projeto de ML dos alunos com uma avaliação manual, você acha que isso reduzirá sua carga de trabalho?

Sim	Não	Sem resposta
✔	⊘	○

Escala Likert de 5 pontos (1 - Concordo totalmente, 2 - Concordo, 3 - Não concordo nem discordo; 4 - Discordo; 5 - Discordo totalmente)

	Concordo totalmente	Concordo	Não concordo nem discordo	Discordo	Discordo totalmente	Sem resposta
Acho que gostaria de usar este sistema com frequência.	○	○	○	○	○	●
Achei o sistema desnecessariamente complexo.	○	○	○	○	○	●
Achei o sistema fácil de usar.	○	○	○	○	○	●
Acho que precisaria do apoio de um técnico para poder usar este sistema.	○	○	○	○	○	●
Achei que as várias funções deste sistema estavam bem integradas.	○	○	○	○	○	●
Achei que havia muita inconsistência neste sistema.	○	○	○	○	○	●
Eu imagino que a maioria das pessoas aprenderia a usar esse sistema muito rapidamente.	○	○	○	○	○	●
Achei o sistema muito complicado de usar.	○	○	○	○	○	●
Eu me senti muito confiante ao usar o sistema.	○	○	○	○	○	●
Eu precisava aprender muitas coisas antes de começar a usar este sistema.	○	○	○	○	○	●

Você acha que a ferramenta de avaliação possui elementos ambíguos ou difíceis de entender?

Escolha uma das seguintes respostas:

- ( ) Sim (se sim, qual?)
- ( ) Não
- (•) Sem resposta

Por favor, coloque aqui o seu comentário:

O que você mais gostou na ferramenta de avaliação?

O que você menos gostou na ferramenta de avaliação?

Mais alguma sugestão?

APÊNDICE 3 - RESULTADOS DA AVALIAÇÃO

Você atua como: [Professor(a) no ensino médio]	Não	Sim	Não	Não	Não	Sim	Não
Você atua como: [Professor(a) na graduação]	Não	Sim	Sim	Não	Não	Sim	Não
Você atua como: [Professor(a) na graduação]	Não	Sim	Sim	Não	Não	Não	Não
Você atua como: [Aluno(a) na graduação]	Não	Não	Não	Sim	Sim	Não	Sim
Você atua como: [Aluno(a) na pós-graduação]	Sim	Não	Não	Não	Não	Sim	Não
Você já desenvolveu quantos modelos de ML para classificação de imagens com Jupyter Notebook?	3-5	Mais do que 5	Mais do que 5	Mais do que 5	2	1	1
Aos professores(as) ensinando ML nas suas disciplinas: como você atualmente avalia os resultados de trabalhos práticos voltados a criação de modelos para classificação de imagens?	NULL	Satisfatório. Os alunos a partir de uma base, caso não consigam programar determinados trechos, conseguem realizar consultas de trechos de códigos que precisam na internet ao desenvolver classificadores.	NULL	NULL	NULL	Atualmente não ensino ML, por isso não posso responder.	Acredito que os trabalhos são interessantes, mas o assunto poderia ser mais aprofundado.
Você conseguiu efetuar o treinamento do modelo no Google Colab?	Sim	Sim	Sim	Sim	Sim	Sim	Sim
[Acho a ferramenta de avaliação útil na educação de computação no ensino médio.]	Concordo	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo	Concordo totalmente
[Acho a ferramenta de avaliação útil no ensino de computação para iniciantes em cursos de	Concordo totalmente	Concordo	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo	Concordo

graduação.]							
[Acho a ferramenta de avaliação útil na educação de computação para iniciantes em cursos de pós-graduação.]	Concordo totalmente	Concordo	Concordo	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo
[Acho que em sua forma atual a ferramenta de avaliação pode ser aplicada de forma prática em minhas aulas.]	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo totalmente	Concordo	Não concordo nem discordo	Não concordo nem discordo
Você acha que existem aspectos/critérios para avaliar a construção de modelos de ML para Classificação de Imagens no contexto de ensino de ML no ensino médio que não são suportados pela ferramenta?	Não	Não	Não	Não	Não	Não	Não
Você acha que existem aspectos/critérios para avaliar a construção de modelos de ML para Classificação de Imagens no contexto de ensino de ML no ensino médio que não são suportados pela ferramenta? [Comentário]	NULL	NULL	NULL	NULL	NULL	NULL	NULL
Você acha que existem aspectos relevantes no que diz respeito ao processo de avaliação da construção de modelos de ML para Classificação de Imagens no contexto de ensino de ML no ensino médio que não são apoiados pela ferramenta de avaliação?	Não	Não	Não	NULL	Não	Não	Não
Você acha que existem aspectos relevantes no	NULL	NULL	NULL	NULL	NULL	NULL	NULL

que diz respeito ao processo de avaliação da construção de modelos de ML para Classificação de Imagens no contexto de ensino de ML no ensino médio que não são apoiados pela ferramenta de avaliação? [Comentário]							
Você acha que as informações de feedback fornecidas são suficientes?	Não (se não, o que está faltando?)	Sim	Sim	Não (se não, o que está faltando?)	Não (se não, o que está faltando?)	Sim	Sim
Você acha que as informações de feedback fornecidas são suficientes? [Comentário]	A avaliação final não deixa claro que tipo de erro eu cometi e como posso melhorar.	NULL	NULL	Algumas métricas como quantidade de imagens achei fácil de entender mas as outras métricas não ficou claro (ou pelo menos eu não entendi) o motivo da pontuação, onde eu "errei" ou como melhorar	Na seção de avaliação de imagens antiéticas: O Instagram censura mamilos femininos. A imagem de uma mulher amamentando seria censurada, mas seria isso antiético? Talvez fosse legal alterar o texto para "imagens não permitidas" para evitar entrar nesse tipo de discussão. Eu tive que reiniciar o ambiente para poder aprimorar a performance. Não sei se isso é normal, mas seria legal ter uma mensagem de erro padrão, nesse caso, que estimula o usuário a reiniciar o ambiente e tentar de novo em caso de erro.	NULL	NULL
Você notou algum erro em relação à funcionalidade da ferramenta de avaliação?	Sim (se sim, qual?)	Não	Sim (se sim, qual?)	Sim (se sim, qual?)	Sim (se sim, qual?)	Não	Sim (se sim, qual?)
Você notou algum erro em relação à funcionalidade	A célula para ver se imagem foi carregada	NULL	Na interpretação da matriz de confusão, ao se clicar	O exportar modelo ONNX resultou em erro	Eu tive que reiniciar o ambiente para alterar o	Algumas células tiveram erros na minha primeira	Na célula final, ocorreu um erro relacionado às

da ferramenta de avaliação? [Comentário]	corretamente às vezes apresentava erro de execução. Tive que executá-la algumas vezes para carregar as imagens.		no título de um acordion já aberto, gera mensagem de erro: “TypeError: list indices must be integers or slices, not NoneType”.	"RuntimeError: No CUDA GPUs are available" mesmo tendo alterado o ambiente para GPU	modelo. Na seção 34, o accordion joga um erro se você clica na opção para fecha-la. Isso pode causar confusão.	tentativa, mas não sei se pode ter sido devido a um conflito de 2 pessoas executando o colab ao mesmo tempo. Na terceira tentativa foi tudo certo :)	acurácias.
Você achou os resultados da avaliação corretos?	NULL	Sim	Não (se não, explique)	Não (se não, explique)	Sim	Sim	Sim
Você achou os resultados da avaliação corretos? [Comentário]	Não sei dizer o que eu errei.	NULL	1. Fiquei em dúvida quanto à totalização da interpretação da primeira Matriz de Confusão. 2. Não consegui entender a totalização da avaliação da acurácia. Só observei o resultado da avaliação como correto quando informei todos os valores como 1.	Por não ter mais detalhes fique confuso com o resultado das avaliações principalmente nos itens 6 e 7	NULL	Só comentando aqui que na primeira tentativa o feedback não apareceu, não sei se pode ter sido devido a um conflito de 2 pessoas executando ao mesmo tempo.	NULL
O desempenho em termos de tempo de processamento da ferramenta de avaliação é aceitável (não leva muito tempo)?	Sim	Sim	Sim	Não	Sim	Sim	Sim
Você conseguiu concluir uma avaliação usando a ferramenta de avaliação?	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Você considera que o tempo que leva para interagir com a ferramenta de avaliação para obter feedback é adequado (não leva muito tempo)?	Sim	Sim	Sim	Não	Sim	Sim	Sim
Comparando o esforço/tempo que você levou para avaliar o projeto de ML dos alunos com uma avaliação manual, você acha que isso reduzirá sua carga de trabalho?	Sim	Sim	Sim	Sim	Sim	N/A	Sim

[Acho que gostaria de usar este sistema com frequência.]	Concordo	Concordo totalmente	Concordo totalmente	Não concordo nem discordo	Concordo	Concordo	Discordo
[Achei o sistema desnecessariamente complexo.]	Discordo totalmente	Discordo totalmente	Discordo	Não concordo nem discordo	Discordo	Discordo	Discordo
[Achei o sistema fácil de usar.]	Concordo	Concordo totalmente	Concordo totalmente	Concordo	Não concordo nem discordo	Concordo totalmente	Concordo
[Acho que precisaria do apoio de um técnico para poder usar este sistema.]	Discordo totalmente	Não concordo nem discordo	Discordo totalmente	Concordo	Concordo	Discordo totalmente	Discordo
[Achei que as várias funções deste sistema estavam bem integradas.]	Concordo totalmente	Concordo totalmente	Concordo totalmente	Discordo	Concordo	Concordo totalmente	Concordo
[Achei que havia muita inconsistência neste sistema.]	Discordo totalmente	Discordo	Discordo totalmente	Não concordo nem discordo	Discordo	Discordo totalmente	Discordo
[Eu imagino que a maioria das pessoas aprenderia a usar esse sistema muito rapidamente.]	Não concordo nem discordo	Concordo totalmente	Concordo	Não concordo nem discordo	Concordo	Não concordo nem discordo	Concordo
[Achei o sistema muito complicado de usar.]	Discordo totalmente	Discordo totalmente	Discordo	Não concordo nem discordo	Discordo	Discordo totalmente	Discordo totalmente
[Eu me senti muito confiante ao usar o sistema.]	Concordo	Concordo totalmente	Concordo totalmente	Não concordo nem discordo	Não concordo nem discordo	Não concordo nem discordo	Não concordo nem discordo
[Eu precisava aprender muitas coisas antes de começar a usar este sistema.]	Discordo totalmente	Discordo totalmente	Discordo totalmente	Concordo	Discordo	Discordo totalmente	Discordo
Você acha que a ferramenta de avaliação possui elementos ambíguos ou difíceis de entender?	Não	NULL	Não	Sim (se sim, qual?)	Não	Não	Não
Você acha que a ferramenta de avaliação possui elementos ambíguos ou difíceis de	NULL	NULL	NULL	Na funcao print_classification_report( ) temos as metricas/colunas	NULL	A ferramenta em si não, mas alguns conceitos de ML/DL podem ser. Talvez fosse interessante uma	NULL

entender? [Comentário]				precision, recall, f1-score, support para cada categoria, mas uma linha para accuracy. Na próxima celula é pedido a acurácia por categoria, fiquei confuso.		explicação conceitual dos mesmos.	
O que você mais gostou na ferramenta de avaliação?	A possibilidade de poder ajustar os parâmetros do modelo, de ver o modelo funcionando para fazer a classificação, e de poder ter uma avaliação rapidamente ao final do processo.	Fácil de entender	O que eu mais gostei foi a apresentação dos resultados da avaliação, de forma da figura do ninja e a tabela com os critérios de avaliação.	Um score final compostos pelas avaliações individuais é muito legal, acho que com mais detalhes vai ajudar muito no processo de use-modify-create.	É direto ao ponto e quando finalizo e vejo meu desempenho, fica ainda mais claro os processos de treinamento.	Ficou muito legal todo o processo de criação e avaliação do modelo dentro de um colab. Em geral, está muito bem organizado. Parabéns pelo trabalho!	Facilidade de execução.
O que você menos gostou na ferramenta de avaliação?	A avaliação não deixa claro que erros foram cometidos.	Não há o que não tenha gostado	Achei que o código necessário para criar os formulários de avaliação e para popular o dicionário pode "poluir" um pouco o código da aplicação de ML no Jupyter.	NULL	Alguns conteúdos podem ser melhorados, como mensagens de erro e títulos.	Da ferramenta nada, só não gostei mesmo das minhas primeiras tentativas que não rodou tudo certinho, mas isso, em si, não tem a ver com a ferramenta :)	Um pouco extensa.
Mais alguma sugestão?	NULL	NULL	- Na análise da Matriz de Confusão, o uso de acordeons pode não deixar claro para o aluno que é necessário clicar em cada um para selecionar a resposta - Na avaliação da acurácia, seria possível exibir no formulário as categorias na ordem em que aparecem na tabela exibida pelo interp.print_classification_report() ? - Seria interessante ocultar todos os códigos das interfaces gráficas por padrão. - Seria interessante também, limpar todas as	NULL	NULL	NULL	NULL

			saídas do Jupyter logo ao iniciar uma nova execução. - Uma sugestão quanto à programação: talvez ficasse mais legal se o jupic já fosse instalado desde o início e, ao invés de ir inserindo os valores em um dicionário, fossem setados atributos de um objeto do jupic. Isso evitaria a necessidade de todo o código para popular o dicionário que acaba “poluindo” o código do modelo. - Os formulários de avaliação poderiam ser gerados dinamicamente pela própria biblioteca jupic, sem a necessidade de inserir o código dos formulários diretamente no Jupyter?				

Desenvolvimento de um Modelo de Avaliação Automatizada de Aprendizagem de *Machine Learning* voltado a Classificação de Imagens no Ensino Médio

Gustavo de Castro Salvador[1]

[1]Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

gustavo.castro.salvador@grad.ufsc.br

Abstract. *Machine Learning (ML) is becoming increasingly present in reality. Although there are already instructional units for teaching ML, the assessment of learning ML concepts remains an open issue and little addressed in existing works. This article presents a student learning assessment model, focusing on high school students for the image classification task. The conceptual model is created following the Evidence-Centered Assessment Design method and automated in the context of creating ML models using Jupyter Notebook in Google Colab environment with Python. A preliminary evaluation indicated the usefulness, functionality and usability of the model.*

Resumo. *Machine Learning (ML) está se tornando cada vez mais presente na realidade. Embora já existam unidades instrucionais para o ensino de ML, a avaliação da aprendizagem dos conceitos de ML continua sendo uma questão aberta e pouco abordada nos trabalhos existentes. Este artigo apresenta um modelo de avaliação de aprendizagem de estudantes, com foco no público do Ensino Médio para classificação de imagens. O modelo conceitual é criado seguindo o método de Evidence-Centered Assessment Design e automatizado no contexto de criação de modelos de ML usando Jupyter Notebook no ambiente do Google Colab com Python. Uma avaliação preliminar indicou a utilidade, funcionalidade e usabilidade do modelo.*

1. Introdução

Aprendizado de máquina ou *Machine Learning* (ML) tornou-se parte da vida cotidiana, impactando profundamente nossa sociedade. No entanto, a maioria das pessoas não entende a tecnologia por trás disso, ofuscando seu potencial de impactar positivamente a sociedade [Evangelista et al., 2018][Ho e Scadding, 2019]. Para desmistificar o ML é importante introduzir conceitos e práticas básicas já na escola, permitindo que os estudantes se tornem não apenas consumidores, mas também criadores de soluções inteligentes [Touretzky et al., 2019][Kandlhofer et al., 2016].

Machine Learning é uma disciplina que estuda o uso de algoritmos e modelos que simulam atividades de aprendizagem humana e se autoaperfeiçoam, obtendo novos conhecimentos e habilidades [Wang et al., 2009]. No entanto, recentemente, algumas iniciativas e projetos que buscam ensinar ML na Educação Básica surgiram. Isto inclui diretrizes curriculares, p.ex., pela AI4K12 [Touretzky et al., 2019] e unidades instrucionais, principalmente extracurriculares [Marques et al., 2020]. Essas unidades

instrucionais ensinam competências que vão de apresentações sobre o que é ML a técnicas específicas da área, com ênfase em redes neurais artificiais e impactos do ML até a aplicação e criação de modelos de Deep Learning. Poucas unidades instrucionais abordam o processo de ML de forma mais completa, com a maioria abordando somente uma parte, como preparação de dados, ou apresentação de alguns processos apenas de forma abstrata, por exemplo, o treinamento de redes neurais [Marques et al., 2020]. Tipicamente são adotadas ferramentas visuais e/ou ambientes textuais como Jupyter Notebooks, às vezes de forma integrada com ambientes de programação baseado em blocos, como Scratch, SNAP! ou App Inventor [Gresse von Wangenheim et al., 2020]. As unidades instrucionais variam também de unidades escolares e cursos no modo presencial até cursos e tutoriais on-line.

As aplicações de *Machine Learning* requerem um grande volume de dados e capacidade de processamento, o que torna a questão de ensino ainda mais complexa: a configuração, hardware e suporte de TI necessário para replicar um ambiente pode torná-la inviável para uma aula com uma quantidade razoável de alunos. Uma ferramenta que soluciona essa questão é o Jupyter Notebook, antigo IPython Notebook, hoje um padrão na comunidade de cientistas de dados [Perkel, 2018].

Como parte do processo de ensino, é importante avaliar a aprendizagem dos estudantes fornecendo feedback tanto para o aluno quanto para o professor [Hattie e Timperley, 2007]. No entanto, apesar de pesquisas existentes para avaliação do ensino de pensamento computacional e programação na Educação Básica [Tang et al., 2019][Lye e Koh, 2014][Oliveira e Oliveira, 2014] observa-se a falta de propostas de avaliações da aprendizagem de ML do aluno [Marques et al., 2020] como revisões sistemáticas, como p.ex. Camada e Durães (2020), Sanusi e Oyelere (2020), Zhou et al. (2020) não analisam de forma explícita esta questão.

2. Metodologia

A metodologia de pesquisa utilizada neste artigo é dividida em cinco etapas principais. A primeira etapa foi a fundamentação teórica, que consiste em estudar, analisar e sintetizar os conceitos principais e a teoria referente aos temas abordados em detalhes em Salvador (2021). Na segunda etapa foi levantado o estado da arte. Nessa etapa é realizado um mapeamento sistemático de literatura seguindo o processo proposto por Petersen et al. (2008) para identificar quais modelos existem para avaliação de aprendizagem de *Machine Learning* do aluno no Ensino Médio. Na terceira etapa, foi realizado o desenvolvimento do modelo de avaliação da aprendizagem seguindo a metodologia de design instrucional ADDIE [Branch, 2009] e o *Evidence-Centered Design* [Mislevy et al., 2003]. Na quarta etapa foi desenvolvida a automação em várias iterações para cada um dos critérios definidos, com base no processo iterativo incremental de engenharia de software [Larman e Basili, 2003]. Por fim, na quinta etapa a rubrica foi realizada uma avaliação preliminar com o objetivo de analisar a qualidade da ferramenta desenvolvida em termos de utilidade, adequação funcional, eficiência de desempenho e usabilidade.

3. Estado da Arte

Com a intenção de identificar e estudar modelos de avaliação de aprendizagem de *Machine Learning* voltado para estudantes do Ensino Médio atualmente sendo utilizados, foi realizado o mapeamento sistemático de literatura seguindo o procedimento proposto por Petersen et al. (2008). O objetivo é responder a seguinte pergunta de pesquisa: Quais modelos existem para avaliação de aprendizagem de *Machine Learning* no Ensino Médio?

As buscas foram realizadas nas principais bases de dados e bibliotecas digitais do campo da Computação: ACM Digital Library, IEEE Xplore, ScienceDirect, SpringerLink, Scopus, arXiv e SocArXiv. Com o intuito de cobrir uma gama maior de publicações, também foram conduzidas pesquisas no Google Scholar, que indexa um grande conjunto de dados de diversas fontes de produção científicas distintas [Haddaway et al. 2015]. A pesquisa no Google também foi utilizada para complementar a pesquisa, minimizando o risco de omitir algum modelo não publicado como artigo científico [Piasecki et al., 2018]. Analisando os resultados das buscas, foram encontradas 12 avaliações de aprendizagem que satisfazem os critérios de inclusão e exclusão.

Foi observado que atualmente a maioria das unidades instrucionais de ensino de *Machine Learning* para estudantes do Ensino Médio não apresentam propostas para a avaliação de aprendizagem dos alunos. Isto pode ser explicado pelo fato que o ensino de ML no Ensino Médio ainda está emergente com um aumento de unidades instrucionais principalmente a partir dos últimos dois anos [Marques et al., 2020]. O foco da maioria dos cursos no nível iniciante explica a ênfase em níveis de aprendizagem mais baixos segundo a Taxonomia de Bloom, com poucos que levam e avaliam a competência ao nível de aplicação de ML. Assim, a maioria das avaliações foca em conceitos básicos a serem avaliados por meio de quizzes e testes. Foram identificados poucos exemplos de rubricas de avaliação de desempenho que analisam artefatos criados pelo aluno ao aplicar conceitos de ML. Outro fator que impede a avaliação de um escopo maior é a típica curta duração destes cursos, muitas vezes de forma extracurricular. O tipo de feedback apresentado para os alunos costuma ser simples, o qual indica se a tarefa foi concluída ou realizada corretamente. A maioria das avaliações propostas são realizadas manualmente pelos instrutores e/ou juízes. Identificou-se a automação de avaliações somente no caso de cursos online referente a quizzes.

Com base nestes resultados é evidente a necessidade de aprimoramento de modelos de avaliação do ensino de ML no Ensino Médio, tanto em termos do seu desenvolvimento e validação sistemático quanto uma cobertura maior dos objetivos de aprendizagem e automação para preparar uma futura adoção mais ampla em escolas brasileiras.

4. Modelo de avaliação de aprendizagem

Visando solucionar os pontos em aberto verificados pelo levantamento do Estado da Arte, foi desenvolvido um modelo de avaliação de aprendizagem. O processo de construção da avaliação é feito com base no *Evidence-Centered Desig*n, uma maneira sistemática de construir avaliações que visa garantir que a evidência coletada e

interpretada é consistente com o conhecimento que a avaliação busca avaliar [Mislevy et al., 2003]. Para isso, foi definido um plano especificando as atividades a serem realizadas por meio de modelos de medição incluindo os modelos de estudante, de evidência e de tarefa.

4.1. Modelo de estudante

O modelo de estudante define uma ou mais variáveis relacionadas ao conhecimento sendo avaliado [Mislevy et al., 2003], servindo de definição para o que se deseja avaliar. No presente trabalho, as variáveis são os objetivos de aprendizagem esperados que os alunos atinjam durante o curso de *Machine Learning*, apresentados na Tabela 1.

Tabela 1. Objetivos de aprendizagem de ML no Ensino Médio

Categoria	Objetivo de Aprendizagem	Nível de aprendizagem com base na Taxonomia Bloom	Referência(s)
OA1. Conceitos básicos de ML	Identificar exemplos de *Machine Learning* e diferenciá-lo da aprendizagem humana	Lembrar	3-A-i K-2, 3-A-i 3-5, 3-A-i 6-8 [AI4K12, 2020]; 1, 2, 3, 5 [Long e Magerko, 2020]
OA2. Redes neurais	Compreender a estrutura de uma rede neural e descrever como suas partes formam um conjunto de funções que computam uma saída capaz de identificar padrões em dados	Compreender	3-A-ii 3-5, 3-B-i 6-8, 3-B-ii 3-5 [AI4K12, 2020]
OA3. Gerenciamento de dados	Preparar um conjunto de dados usado para treinar um modelo de ML considerando o tamanho do conjunto de dados, a forma com que os dados foram coletados e rotulados, além de sua qualidade (equilíbrio, balanceamento, viés)	Aplicar	3-C-ii 9-12 [AI4K12, 2020]; 11, 12 [Long e Magerko, 2020]; 1A-DA-05 [CSTA, 2017]
OA4. Treinamento de modelo de ML	Treinar um modelo de ML para classificação/predição usando um algoritmo de aprendizagem supervisionada com dados reais e ajustando os parâmetros de treinamento	Aplicar	3-A-ii 9-12, 3-A-iii 9-12 [AI4K12, 2020]
OA5. Avaliação e interpretação do desempenho de um modelo de ML	Analisar e interpretar o desempenho de um modelo de ML para classificação/predição	Aplicar	(AMERSHI et al., 2019)
OA6. Implantação de um modelo de ML	Exportar um modelo e integrar o modelo dentro de um sistema de software	Aplicar	(AMERSHI et al., 2019)
OA7. Processo de ML	Compreender e aplicar as etapas envolvidas no *Machine Learning* e suas práticas e desafios	Aplicar	3-A-iv 9-12 [AI4K12, 2020]; 9 [Long e Magerko, 2020]
OA8. Ética de ML	Identificar e descrever diferentes questões éticas acerca de ML (privacidade, viés introduzido por características dos dados de treinamento, tomadas de decisões éticas, etc.)	Compreender	3-C-iii 6-8 [AI4K12, 2020]; 3A-AP-24 [CSTA, 2017]; 13, 16 [Long e Magerko, 2020]
OA9. Impactos do IA/ML	Identificar prós e contras de IA e ML para atividades cotidianas e opções de carreira atuais e futuras	Compreender	2-IC-21 [CSTA, 2017]; 6 [Long e Magerko, 2020]
OA10. Criar/modificar programas	Criar programas de computador usando sequências, eventos e outros comandos ou modificar programas existentes	Aplicar	1B-AP-10, 1B-AP-12 [CSTA, 2017]
OA11. Testar e aperfeiçoar programas	Testar e aperfeiçoar artefatos computacionais	Aplicar	1B-AP-15, 2-DA-09, 3A-IC-25 [CSTA, 2017]
OA12. Botânica	Reconhecer espécies de árvores nativas de Santa Catarina	Lembrar	[Ministério da Educação, 2017]

4.2. Modelo de evidência

O modelo de evidência é baseado em comportamentos ou produtos observáveis resultantes de respostas a uma tarefa específica [Zieky, 2014], apresentando como os objetivos mensurados devem ser avaliados de acordo com determinado desempenho atingido. Para realizar a avaliação de desempenho a partir dos artefatos produzidos pelos estudantes durante o treinamento do modelo, é definida uma rubrica de avaliação apresentada na Tabela 2.

Tabela 2. Rubrica de avaliação

		Níveis de desempenho		
ID	**Critério**	***Baixo - 0 pt.***	***Aceitável - 1 pt.***	***Bom - 2 pt.***
Preparação de dados (OA3)				
C1	Quantidade de imagens	Menos de 5 imagens por categoria	6 de 10 imagens por categoria	Mais de 10 imagens por categoria
C2	Distribuição do conjunto de dados	Quantidade de imagens por categoria varia muito	Quantidade de imagens por categoria varia pouco	Todas as categorias possuem a mesma quantidade de imagens
Preparação de dados/Botânica (OA3/OA12)				
C3	Rotulagem das imagens	Menos de 20% das imagens rotuladas corretamente	De 20% a 99% das imagens rotuladas corretamente	Todas as imagens rotuladas corretamente
Treinamento de modelo de ML/Transfer Learning e Fine-Tuning (OA4)				
C4	Treinamento - Transfer Learning	O modelo não foi treinado (transfer learned)	O modelo foi treinado com os parâmetros padrão	O modelo foi treinado com parâmetros ajustados (arquitetura, época e taxa de aprendizagem)
C5	Treinamento - Fine-Tuning	O modelo não foi fine-tuned	Foi feito unfreeze das camadas e melhor taxa de aprendizagem não encontrada ou modelo não treinado	Foi feito unfreeze das camadas, a melhor taxa de aprendizagem foi encontrada e o modelo foi fine-tuned
Avaliação e interpretação do desempenho de um modelo de ML (Transfer Learning e Fine-Tuning) (OA5)				
C6	Interpretação de acurácia	Categorias com baixa acurácia não identificadas	Categorias com baixa acurácia identificadas e interpretação incorreta em relação ao modelo	Categorias com baixa acurácia identificadas corretamente e interpretação correta em relação ao modelo
C7	Interpretação da matriz de confusão	Classificações incorretas não identificadas e interpretação incorreta em relação ao modelo	Classificações incorretas identificadas e interpretação incorreta em relação ao modelo	Classificações incorretas identificadas e interpretação correta em relação ao modelo
C8	Ajustes/melhorias feitas	Sem novas iterações de desenvolvimento	Uma nova iteração com alterações no conjunto de dados e/ou parâmetros de treinamento	Diversas novas iterações com alterações no conjunto de dados e/ou parâmetros de treinamento
Avaliação e interpretação do desempenho de um modelo de ML/Testar e aperfeiçoar programas (OA5/OA11)				
C9	Testes com novos objetos	Nenhum novo objeto testado	1-2 novos objetos testados	Mais de dois novos objetos testados
C10	Interpretação dos testes	Interpretação errada	---	Interpretação correta

Para o modelo de medição, é feito um cálculo para chegar a uma nota final para as atividades realizadas pelo estudante no curso, sendo a soma da pontuação dos 10 critérios apresentados divididos por 2.

4.3. Modelo de tarefa

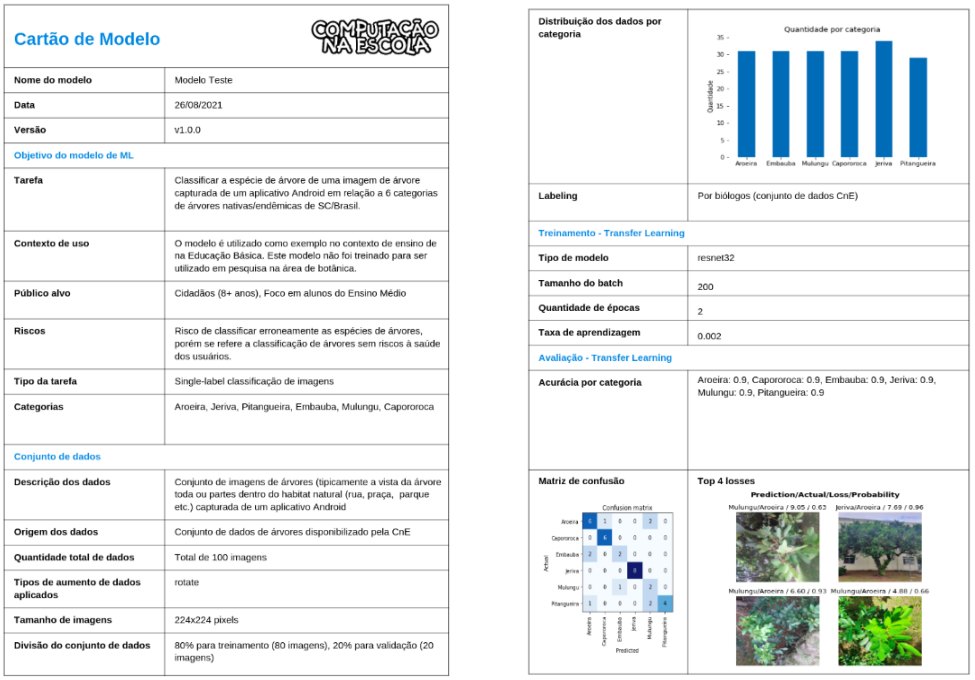
O modelo de tarefa descreve o material que o avaliado produzirá, determinando onde o modelo de avaliação é aplicado. A tarefa do artigo em questão é apresentada na Tabela 3.

Tabela 3. Definição do modelo de tarefa

Objetivo do modelo de DL		
Tarefa	Classificar a espécie de árvore de uma imagem de árvore (tipicamente a vista da árvore toda ou partes dentro do habitat natural (rua, praça, parque etc.) capturada de um aplicativo Android em relação a 6 categorias de árvores nativas/endêmicas de SC/Brasil.	
Tipo da tarefa	Single-label classificação de imagens	
Categorias	6 categorias de espécies de árvores nativas/endêmicas de SC/Brasil	Aroeira-vermelha, Capororoca, Embaúba, Jerivá, Mulungu, Pitangueira
Experiência	Conjunto de imagens de árvores (tipicamente a vista da árvore toda ou partes dentro do habitat natural (rua, praça, parque etc.)	
Fonte de dados	Conjunto de dados de árvores disponibilizado pela CnE	
Quantidade de dados	No total 215 imagens	
Padronização das imagens	Formato: .jpeg	Tamanho: 224x224 pixels
Rotulação de dados	A ser feito pelos estudantes do ensino médio	
Desempenho	O modelo de ML será otimizado para precisão para reduzir o risco de indicar a espécie errada ao usuário, levando ele a uma compreensão errada	
	Medido pela precisão e acurácia de no mínimo 0.75 total e por categoria de espécie de uma imagem em relação ao valor verdadeiro definido por biólogos	
Medidas	Acurácia (total/por categoria)	No mínimo 0.75
	Precisão	No mínimo 0.75

4.4. Modelo de documentação

Com o objetivo de automatizar a documentação das principais características do modelo de ML criado pelo estudante é especificado um Cartão de Modelo com base em propostas similares [Mitchell et al., 2019]. Ele centraliza informações sobre um modelo de ML treinado, p.ex., como foi construído, quais suposições foram feitas durante seu desenvolvimento, quais algoritmos foram utilizados, quais as características do conjunto de dados utilizados, etc., como apresentado nos exemplos da Figura 1.



Cartão de Modelo

Nome do modelo	Modelo Teste
Data	26/08/2021
Versão	v1.0.0
Objetivo do modelo de ML	
Tarefa	Classificar a espécie de árvore de uma imagem de árvore capturada de um aplicativo Android em relação a 6 categorias de árvores nativas/endêmicas de SC/Brasil.
Contexto de uso	O modelo é utilizado como exemplo no contexto de ensino de na Educação Básica. Este modelo não foi treinado para ser utilizado em pesquisa na área de botânica.
Público alvo	Cidadãos (8+ anos), Foco em alunos do Ensino Médio
Riscos	Risco de classificar erroneamente as espécies de árvores, porém se refere a classificação de árvores sem riscos à saúde dos usuários.
Tipo da tarefa	Single-label classificação de imagens
Categorias	Aroeira, Jeriva, Pitangueira, Embauba, Mulungu, Caporoca
Conjunto de dados	
Descrição dos dados	Conjunto de imagens de árvores (tipicamente a vista da árvore toda ou partes dentro do habitat natural (rua, praça, parque etc.) capturada de um aplicativo Android
Origem dos dados	Conjunto de dados de árvores disponibilizado pela CnE
Quantidade total de dados	Total de 100 imagens
Tipos de aumento de dados aplicados	rotate
Tamanho de imagens	224x224 pixels
Divisão do conjunto de dados	80% para treinamento (80 imagens), 20% para validação (20 imagens)

Distribuição dos dados por categoria	
Labeling	Por biólogos (conjunto de dados CnE)
Treinamento - Transfer Learning	
Tipo de modelo	resnet32
Tamanho do batch	200
Quantidade de épocas	2
Taxa de aprendizagem	0.002
Avaliação - Transfer Learning	
Acurácia por categoria	Aroeira: 0.9, Caporoca: 0.9, Embauba: 0.9, Jeriva: 0.9, Mulungu: 0.9, Pitangueira: 0.9
Matriz de confusão	
Top 4 losses	

Figura 1. Exemplo de modelo de documentação

5. Automação da avaliação de aprendizagem

A partir da análise e modelagem do domínio e do framework conceitual desenvolvido, foram realizadas várias iterações para automatizar a rubrica de avaliação. A automação da avaliação implementada neste trabalho na forma de um pacote Python, chamado jupiclass (**Jup**yter Notebook **I**mage **Cl**assification **ass**essment).

As informações sobre o conjunto de dados, modelo, treinamento do *Transfer Learning* e *Fine-Tuning* e respostas das questões de interpretação devem ser fornecidas em um objeto Python da classe *ImageClassification*. Esse objeto deve estar disponível e com seus atributos preenchidos no código Python do Jupyter Notebook em que será aplicada a automação da rubrica. A partir desse objeto, é possível realizar a avaliação do treinamento seguindo a rubrica proposta no modelo de evidência, gerar um PDF seguindo o modelo de documentação e apresentar visualmente os resultados da avaliação com um mascote "ninja-robô" (Figura 2). O pacote está disponível no repositório PyPI, conforme apresentado em http://computacaonaescola.paginas.ufsc.br/jupiclass/. A rubrica de avaliação foi aplicada e testada em um Jupyter Notebook do curso para a classificação de imagens.

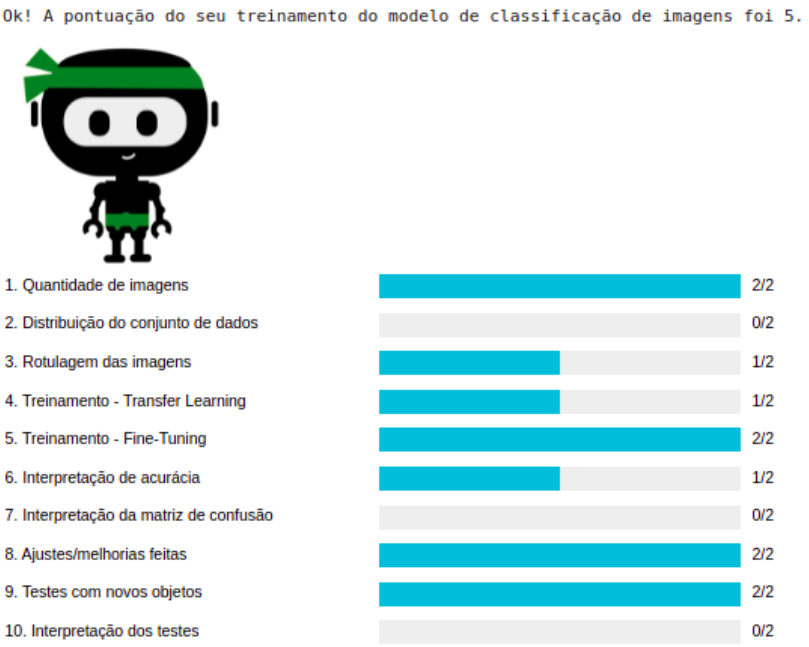


Figura 2. Exemplo de apresentação da avaliação para a nota 5

O módulo foi testado conforme a corretude, para assegurar o possível que a implementação feita corresponde ao proposto pela rubrica.

6. Avaliação do modelo de aprendizagem

Para avaliar a qualidade do modelo de avaliação, foi realizada uma avaliação preliminar com o objetivo de analisar a qualidade da ferramenta desenvolvida em termos de utilidade, adequação funcional, eficiência de desempenho e usabilidade do ponto de vista de professores e alunos no contexto do ensino de *Machine Learning*.

A avaliação da ferramenta foi realizada em Agosto de 2021 por um total de 14 participantes, incluindo professores do Ensino Médio e ensino superior e alunos do Ensino Superior em Santa Catarina/Brasil. Foram convidados no total 23 participantes via e-mail, dos quais 14 responderam a pesquisa representando uma taxa de resposta de 60%. Destas 14 respostas, foram 7 completas.

6.1. Análise

Todos os participantes consideraram a ferramenta de avaliação útil no ensino de ML no Ensino Médio, assim como para o ensino de iniciantes em cursos de graduação e pós-graduação. Além disso, a maioria dos professores consideraram que a ferramenta em seu estado atual pode ser aplicada de forma prática em suas aulas.

Todos os participantes indicaram que a ferramenta suporta todos os aspectos/critérios para avaliar a construção de modelos de ML para classificação de imagens no contexto de ensino de ML no Ensino Médio, assim como aspectos relevantes no que diz respeito ao processo de avaliação. Dois participantes fizeram uma sugestão de incluir quais foram os erros cometidos pelo estudante durante o treinamento do modelo nas informações de feedback apresentadas.

A maioria dos participantes se deparou com erros durante a execução do Colab, decorrentes dos problemas técnicos já relatados de execução simultânea, além de problemas com GPUs do Google Colab e com o widget da questão de interpretação da matriz de confusão. Mensagens de erro mais explicativas poderiam ter guiado melhor os participantes na execução. Embora a maioria dos participantes achou o resultado da avaliação correto, alguns questionaram como o cálculo das questões de interpretação é feito, dado que os critérios avaliados e suas respectivas pontuações não foram apresentados em um primeiro momento. Quase todos os participantes consideraram aceitável o desempenho da ferramenta de avaliação em termos de tempo de processamento.

Todos os participantes que responderam completamente o questionário conseguiram concluir ao final uma avaliação do treinamento do modelo no Google Colab com a ferramenta de avaliação. Eles consideram que o tempo que leva para interagir com a ferramenta de avaliação para obter feedback é adequado. Todos acreditam que a ferramenta reduz a carga de trabalho em comparação com uma avaliação manual de projeto de ML. Aplicando o System Usability Scale (SUS) [Brooke, 1996] para medir a satisfação com a ferramenta, a pontuação média foi de 81.78, indicando um nível de usabilidade aceitável e satisfação muito boa, como apresentado na Tabela 4.

Tabela 4. Resultados da pontuação SUS

	Média	Valor mínimo	Valor máximo
jupiclass	81.78	65	92.5

A apresentação dos resultados, facilidade de entendimento e velocidade foram citados como pontos fortes. Como fracos, a falta de visualização de quais erros foram cometidos e necessidade de melhorias nas mensagens de erro foram apontadas.

6.2. Discussão

Os resultados da avaliação fornecem uma indicação inicial que a ferramenta jupiclass pode ser útil, funcional, eficiente em desempenho e ter boa usabilidade para a avaliação de treinamentos de modelos de ML para a tarefa de Classificação de Imagens de estudantes do Ensino Médio. A apresentação visual dos resultados com o ninja robô e facilidade de uso da ferramenta foram as principais características positivas levantadas. Como foi observado, uma melhor descrição de como as competências são avaliadas e como a pontuação é gerada faz-se necessária para o entendimento do aluno de como seu treinamento está sendo avaliado. Outro aspecto verificado é que uma avaliação mais completa e com menor atrito na execução por parte dos participantes demanda um ambiente mais controlado, isolado e sem interferências para a execução dos Jupyter Notebooks.

Os resultados obtidos nesta avaliação devem ser interpretados com cautela, levando em consideração as ameaças potenciais à sua validade. Devido à falta de medições em um contexto educacional real e/ou com grupo de controle, os resultados são limitados a fornecer apenas uma primeira indicação sobre a qualidade da ferramenta jupiclass. Embora os participantes tenham sido selecionados de forma que seus perfis correspondessem aos usuários em potencial, a falta de mais professores do Ensino

Médio e ausência de estudantes desta faixa pode influenciar nos resultados. Sem integrantes do público alvo, possíveis necessidades de mais explicações podem ter sido relevadas dado o conhecimento prévio dos participantes. Assim, indica-se a necessidade de estudos futuros com maior número de participantes do perfil específicos do público alvo.

7. Conclusão

Este artigo apresenta o desenvolvimento de um modelo de avaliação de aprendizagem de *Machine Learning* voltado à tarefa de classificação de imagens para o contexto do Ensino Médio.

Em termos de impacto científico foi um modelo conceitual para avaliar a aprendizagem de ML no Ensino Médio, definindo uma rubrica inédita conforme o estado da arte levantado. A partir do modelo conceitual, foi criada uma biblioteca automatizando a avaliação com base na rubrica dentro do Jupyter Notebook e gerando um resumo do modelo treinado para fins de documentação. Também foi realizada uma avaliação preliminar da biblioteca criada com um expert panel, gerando resultados iniciais promissores sobre a proposta da ferramenta. Em termos sociais, espera-se realizar uma contribuição importante para o ensino de ML nas escolas Brasileiras, buscando facilitar a avaliação da aprendizagem do aluno e assim contribuir no seu progresso na aprendizagem como um todo.

Referências

AI4K12. (2020) “Draft Big Idea 3 - Progression Chart. 2020”, https://drive.google.com/file/d/1QL6I_I5cdNTVnYBlZ3_Lxur2DgFjmG_d/view, Outubro.

Branch, R. M. (2009) “Instructional Design: The ADDIE Approach”, Springer.

Brooke, J. (1996) “SUS—A Quick and Dirty Usability Scale”, Usability Evaluation in Industry, 189(1), p. 4-7.

Camada, M. Y. e Durães, G. M. (2020) “Ensino da Inteligência Artificial na Educação Básica: um novo horizonte para as pesquisas brasileiras”, Anais do Simpósio Brasileiro de Informática na Educação.

CSTA. (2017) “CSTA K-12 Computer Science Standards”, http://www.csteachers.org/standards, Outubro.

Evangelista, I., Blesio, G. e Benatti, E. (2018) “Why Are We Not Teaching Machine Learning at High School?” A Proposal. Proc. of the World Engineering Education Forum, Albuquerque, NM, USA.

Ho, J. W. K. e Scadding, M. (2019) “Classroom Activities for Teaching Artificial Intelligence to Primary School Students”, In: Proc. of the Int. Conference on Computational Thinking, Hong Kong, China.

Kandlhofer, M. et al. (2016) “Artificial Intelligence and Computer Science in Education: From Kindergarten to University”. In: Proc. of the IEEE Frontiers in Education Conference, Erie, PA, USA.

Gresse von Wangenheim, C., Marques, L. S. e Hauck, J. C. R. (2020) “Machine Learning for All – Introducing Machine Learning in K-12”, SocArXiv.

Haddaway, N. R. et al. (2015) “The Role of Google Scholar in Evidence Reviews and Its Applicability to Grey Literature Searching”, PLOS ONE, 10(9).

Hattie, J. e Timperley, H. (2007) “The power of feedback”, Review of Educational Research, 77(1), p. 81-112.

Larman, C. e Basili, V. R. (2003) “Iterative and Incremental Development: A Brief History”, Computer, 36(6), p. 47-56.

Long, D. e Magerko, B. (2020) “What is AI Literacy? Competencies and Design Considerations”, In: Proc. of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, p. 1-16.

Lye, S. Y. e Koh, J. H. L. (2014) “Review on teaching and learning of computational thinking through programming: What is next for K-12?”, Computers in Human Behavior, 41(1), p. 51–61.

Marques, L. S., Gresse von Wangenheim, C. e Hauck, J. C. R. (2020) “Teaching Machine Learning in School: A Systematic Mapping of the State of the Art”, Informatics in Education, 19(2), p. 283–321.

Ministério da Educação. (2017) “Base Nacional Comum Curricular”, http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf, Outubro.

Mislevy, R. J., Almond, R. G. e Lukas, J. F. (2003) “A Brief Introduction to Evidence-centered Design”, Educational Testing Service, Research & Development Division, Princeton, NJ, USA.

Mitchell, M. et al. (2019) “Model Cards for Model Reporting”, arXiv:1810.03993.

Oliveira, M. e Oliveira, E. (2014) “Metodologia de Diagnóstico e Regulação de Componentes de Habilidades da Aprendizagem de Programação”, Anais do Workshop sobre Educação em Computação, Brasília, Brasil.

Perkel, J. M. (2018) “Why Jupyter is data scientists' computational notebook of choice”, Nature, 563(7729), p. 145-146.

Petersen, K., Feldt, R., Mujtaba, S. e Mattsson, M. (2008) “Systematic Mapping Studies in Software Engineering”, In: Proc. of the 12th Int. Conference on Evaluation and Assessment in Software Engineering, Bari, Italy.

Piasecki, J., Waligora, M. e Dranseika, V. (2018) “Google Search as an Additional Source in Systematic Reviews”, Science and Engineering Ethics, 24(2), p. 809-910.

Salvador, G. C. (2021) “Desenvolvimento de um Modelo de Avaliação Automatizada de Aprendizagem de Machine Learning voltado à Classificação de Imagens no Ensino Médio”, Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Universidade Federal de Santa Catarina, Brasil.

Sanusi, I. T. e Oyelere, S. S. (2020) “Pedagogies of Machine Learning in K-12 Context”, IEEE Frontiers in Education Conference, Uppsala, Sweden.

Tang, D., Utsumi, Y. e Lao, N. (2019) “PIC: A Personal Image Classification Webtool for High School Students”. In: Proc. of the Int. Joint Conferences on Artificial Intelligence EduAI Workshop, Macao, China.

Touretzky, D. S. et al. (2019) “Envisioning AI for K-12: What Should Every Child Know about AI?”, In: Proc. of the 33rd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, Honolulu, HI, USA.

Wang, H. et al. (2009) “A Brief Review of Machine Learning and Its Application”, In: Proc. of the Int. Conference on Information Engineering and Computer Science, Wuhan, China, p. 1-4.

Zhou, X., Brummelen, J. V. e Lin, P. (2020) “Designing AI Learning Experiences for K-12: Emerging Works, Future Opportunities and a Design Framework”, arXiv:2009.10228.

Zieky, M. J. (2014) “An introduction to the use of evidence-centered design in test development”, Psicología Educativa, 20(2), p. 79-87.