

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**Desenvolvimento de uma Ferramenta Visual de Classificação de
Imagens para o Ensino de *Machine Learning* no Ensino Médio**

Augusto César Medeiros Franz

Florianópolis – SC

2021/1

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

**Desenvolvimento de uma Ferramenta Visual de Classificação de
Imagens para o Ensino de *Machine Learning* no Ensino Médio**

Augusto César Medeiros Franz

Trabalho de conclusão de curso
apresentado como parte dos requisitos
para obtenção do grau de Bacharel em
Sistemas de Informações.

Florianópolis

2021/1

Augusto César Medeiros Franz

Desenvolvimento de uma Ferramenta Visual de Classificação de
Imagens para o Ensino de *Machine Learning* no Ensino Médio

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção
do grau de Bacharel em Sistemas de Informações

Orientador: Prof. Dr. Jean Carlo Rossa Hauck

Co-Orientadora: Profa. Dra. rer. nat. Christiane Gresse von Wangenheim, PMP

Banca examinadora:

Prof. Ederson Torresini

SUMÁRIO

1.	INTRODUÇÃO.....	10
1.1.	Contextualização.....	10
1.2.	Objetivos	13
1.3.	Metodologia de Pesquisa	14
1.4.	Estrutura deste Documento.....	15
2.	FUNDAMENTAÇÃO TEÓRICA	17
2.1.	<i>Machine Learning</i>	17
2.2.	Ensino do <i>Machine Learning</i> no Ensino Médio	23
2.3.	Ambientes de programação visuais	28
2.4.	Jupyter Notebook e Google Colab	30
3.	ESTADO DA ARTE	34
3.1.	Definição do protocolo de revisão	34
3.2.	Execução da busca	36
3.3.	Análise dos resultados	38
	3.3.1. Quais ferramentas visuais existem para ensinar ML no ensino médio?	38
	3.3.2. Quais são suas características educacionais?	41
	3.3.3. Quais são suas características em relação à plataforma de ML?	43
	3.3.4. Quais são suas características em relação à plataforma de implantação?	52
	3.3.5. Como as ferramentas foram desenvolvidas e avaliadas?	53
3.4.	Discussão.....	56
4.	<i>Visual Layer Classification</i> para o desenvolvimento de modelo de classificação de imagens.....	59

4.1.	Análise de Requisitos.....	59
4.2.	Arquitetura do sistema	60
4.3.	Implementação do Vulcan.....	62
4.3.1.	Integração com o Google Drive	65
4.3.2.	Biblioteca <i>Fast.ai</i>	66
4.3.3.	Processo de <i>ML</i>	66
4.3.4.	Interface visual	68
4.3.5.	Distribuição.....	74
4.4.	Exemplo de Aplicação: Modelo de ML de reconhecimento de árvores nativas de Santa Catarina.....	75
4.4.1.	Conjunto de dados	75
4.4.2.	Treinamento e desempenho.....	76
5.	Avaliação do Visual Layer Classification	84
5.1.	Definição da avaliação	84
5.2.	Execução da avaliação	88
5.3.	Análise dos resultados	89
5.3.1.	Utilidade	89
5.3.2.	Funcionalidade	90
5.3.3.	Desempenho	91
5.3.4.	Usabilidade	92
5.4.	Discussão.....	94
6.	Conclusão.....	99
	REFERÊNCIAS	101
	Apêndice A - Resultados do survey	109
	Apêndice B - Desenvolvimento de uma Ferramenta Visual de Classificação de Imagens para o Ensino de Machine Learning no Ensino Médio.....	117

LISTA DE FIGURAS

Figura 1 - Exemplo de algoritmo de ML com diferentes tipos de dados de entrada.	18
Figura 2. Exemplo de uma interface textual de um Jupyter Notebook no Google Colab.....	31
Figura 3 - Ambientes com suporte para implantação dos modelos de ML criados...	40
Figura 4 - Exemplos de suporte ao desenvolvimento de ML.....	44
Figura 5 - Frequência de tarefas e tipos de dados de ML compatíveis.....	45
Figura 6 - Frequência das etapas do processo de ML com suporte.....	46
Figura 7 - Exemplos de suporte para avaliação dos modelos de ML.....	51
Figura 8 - Exemplo de suporte de implantação baseado em bloco (PIC).....	52
Figura 9 - Visão geral do Vulcan.....	61
Figura 10 - Etapas do processo de ML centrado no ser humano.....	67
Figura 11 - Exemplo da instalação e execução do Vulcan.....	74
Figura 12 - Visão geral da distribuição demográfica dos participantes.....	89

LISTA DE TABELAS

Tabela 1 - Objetivos de aprendizado para o ensino médio.....	25
Tabela 2 - Termos de Busca e <i>String</i> de pesquisa para cada fonte.....	36
Tabela 3 - Número de artigos identificados por estágio de seleção.....	37
Tabela 4 - Ferramentas visuais para ensinar ML no ensino médio.....	38
Tabela 5 - Características gerais das ferramentas.....	41
Tabela 6 - Características educacionais das ferramentas.....	42
Tabela 7 - Características gerais da plataforma de ML.....	46
Tabela 8 - Características relativas aos dados.....	47
Tabela 9 - Características do modelo de ML e aprendizagem.....	48
Tabela 10 - Características relativas à avaliação.....	50
Tabela 11 - Características da plataforma de implantação.....	53
Tabela 12 - Informações sobre o desenvolvimento das ferramentas.....	53
Tabela 13 - Informações sobre a avaliação das ferramentas.....	55
Tabela 14 - Atividades suportadas pelo Vulcan.....	62
Tabela 15 - Resultado obtido na interface visual.....	68
Tabela 16 - Árvores nativas selecionadas para a criação do modelo.....	75
Tabela 17 - Informações do conjunto de dados.....	75
Tabela 18 - Análise de requisitos para o treinamento do modelo.....	76
Tabela 19 - Resultados do treinamento dos modelos.....	77
Tabela 20 - Visão geral da decomposição das características de qualidade e operacionalização da medição.....	84
Tabela 21 - Análise da utilidade da ferramenta.....	90
Tabela 22 - Análise da funcionalidade da ferramenta.....	91
Tabela 23 - Análise do desempenho da ferramenta.....	92

Tabela 24 - Análise da usabilidade da ferramenta.....	92
Tabela 25 - Análise do tempo de treinamento da ferramenta.....	92
Tabela 26 - Análise da comparação da ferramenta com ambiente tradicional.....	92
Tabela 27 - Análise do questionário SUS.....	93
Tabela 28 - Análise da operabilidade da ferramenta.....	94
Tabela 29 - Comentários dos alunos e professores.....	95

LISTA DE ABREVIATURAS

IA - Inteligência Artificial

ML - *Machine Learning*

SBC - Sociedade Brasileira de Computação

AI4K12 - *AI for K-12 Working Group*

AAAI - *Association for the Advancement of Artificial Intelligence*

CSTA - *Computer Science Teachers Association*

RESUMO

A inteligência artificial já está presente na vida de boa parte da população gerando um grande e crescente impacto tanto econômico quanto social. Assim, é muito importante que os cidadãos não sejam apenas consumidores de inteligência artificial, mas que tenham também o conhecimento mínimo para a utilizar de forma crítica e consciente. Uma forma de difundir e popularizar esse conhecimento pode ser o ensino de conceitos básicos de *Machine Learning* no ensino médio preparando os jovens para o futuro. Com esta motivação, o presente trabalho tem como objetivo geral desenvolver uma ferramenta visual de programação voltada para web (Jupyter) para o ensino de *Machine Learning* no ensino médio. A ferramenta aborda o processo completo desde a preparação de dados até a geração do modelo para exportação em ONNX. Com base na fundamentação teórica e levantamento do estado da arte foi desenvolvida uma interface visual em um *Jupyter notebook* rodando no *Google Colab* voltada a classificação de imagens. Resultados de uma primeira avaliação indicam que a solução criada pode ser útil, funcional, eficiente em desempenho e com boa usabilidade. Assim espera-se que os resultados do presente trabalho possam contribuir para facilitar o ensino de inteligência artificial e *machine learning* no ensino médio em escolas brasileiras.

Palavras chave: Inteligência artificial, *Machine Learning*, Ensino médio, Ambiente visual de programação

1. INTRODUÇÃO

1.1. Contextualização

A Inteligência Artificial (IA) vem ganhando popularidade e está cada vez mais presente na vida das pessoas, gerando um grande impacto social e econômico (SHEIN, 2020), mesmo muitas vezes passando despercebida. Diariamente é utilizado diferentes IA sem nem perceber, como conversar com algum *chatbot*, as sugestões de palavras do teclado do celular e até os assistentes pessoais virtuais. Atividades complexas também fazem uso de IA como a navegação de carros autônomos, soluções de simulações computacionais, algoritmos com a finalidade de descobrir exoplanetas (ADAMS, 2020) e até mesmo soluções para auxiliar na vacina para a COVID-19 (ROBY, 2020).

A IA pode ser caracterizada como a concepção e construção de agentes inteligentes que recebem percepções do ambiente e executam ações que afetam esse ambiente (RUSSEL; NORVIG, 2009). IA representa uma ampla área de conhecimento que pode ser desmembrada em diversas outras, cada uma destas utilizando uma metodologia e/ou abordagem com diferentes finalidades e muitas vezes complementando umas às outras.

Uma destas áreas de conhecimento é o *Machine Learning* (ML), que permite que computadores tenham a capacidade de aprender a partir de exemplos e dados (SAMUEL, 2000). Ao permitir que computadores consigam realizar tarefas específicas de maneira inteligente, os sistemas de ML podem realizar processos complexos a partir de dados, em vez de seguir apenas regras pré-programadas (ROYAL SOCIETY, 2017). Algumas tarefas típicas solucionadas com ML são o reconhecimento de fala e a classificação de imagens (KRIZHEVSKY et al. 2012), aumentando cada dia mais o

seu uso em aplicações de segurança, diagnóstico médico por imagem, entre muitas outras. Com o uso intenso das áreas e tecnologias de IA/ML as mesmas têm se tornado cada dia mais difundidas e importantes na sociedade.

Assim, é importante popularizar o conhecimento mínimo além de meramente consumi-las, podendo assim ser utilizado de forma crítica e consciente (TOURETZKY et al., 2019b)(KANDLHOFER et al., 2016). Uma das formas de difundir o conhecimento de IA/ML é na educação básica, formando um cidadão consciente, visando incluir minorias além de apontar possíveis carreiras profissionais aos jovens nesta área de conhecimento (KAHN et al., 2020). Atualmente já existem países que incluíram em seus currículos escolares o ensino de IA. Um exemplo é a China, que adotou o ensino de IA em todas as escolas de ensino médio (JING, 2018).

Apesar disso, o ensino de IA/ML na educação básica de forma geral ainda só está iniciando. Atualmente estão sendo definidos guias para o ensino da computação na educação básica como o *CSTA K-12 Computer Science Framework* (CSTA, 2017) ou as diretrizes para o ensino de computação na educação básica da Sociedade Brasileira de Computação (SBC, 2020) que até abordam IA, mas não se aprofundando no assunto nem citando abordagens de ML. Com isso alguns grupos como *AI for K-12 Working Group* (AI4K12), *Association for the Advancement of Artificial Intelligence* (AAAI) e *Computer Science Teachers Association* (CSTA) visam criar um guia para ensinar IA/ML na educação básica mirando 5 grandes ideias (TOURETSKY et al., 2019a): percepção, representação e raciocínio, aprendizagem, interação natural e por fim impacto social.

Também já existem várias iniciativas e alguns cursos/tutoriais voltados ao ensino de ML na educação básica (MARQUES et al., 2020), que tipicamente usam ambientes visuais e baseados em blocos neste contexto. Tipicamente nessa faixa

etária para iniciar o ensino de computação se usa linguagens visuais, como Scratch (MIT, 2020) ou o App Inventor (MIT, 2019). As linguagens de programação visuais combinam elementos visuais pré-definidos para desenvolver o código que será executado, limitando assim a possibilidade de erros que comumente aconteceriam em linguagens textuais. Observa-se inclusive uma tendência atual forte para o uso de ambientes visuais (ou chamado de *no-coding*) também para o desenvolvimento de modelos de ML, como p.ex. o Google Teachable Machine (GOOGLE, 2020). Muitos acabam criando um nível de abstração muito grande, em que os alunos acabam não aprendendo o processo de ML criando uma grande “caixa preta” em que os dados apenas entram e saem classificados (HITRON *et al.*, 2019). Por tratar-se de um processo complexo é muito importante encontrar um equilíbrio entre manter algumas etapas nesta “caixa preta” e manter outros abertos a descobertas, encorajando os jovens a conhecer e entender cada vez mais sobre ML (RESNICK *et al.*, 2000).

Porém, como estes ambientes tipicamente são criados especificamente para fins educacionais eles se tornam pouco distantes de ferramentas mais comumente utilizadas na prática. Entre estas ferramentas mais convencionais se destaca atualmente ambientes de programação virtuais como Jupyter Notebooks (JUPYTER, 2021), em que apenas com o navegador é possível desenvolver usando uma linguagem textual de programação (normalmente Python) em diversas atividades, incluindo IA e ML podendo executar treinamento no Google Colab de forma gratuita.

Porém, no nível educacional de ensino médio a aprendizagem de ML com linguagens textuais pode ser difícil para iniciantes (IDREES *et al.*, 2018)(WEINTROP; WILENSKY, 2017). No início as linguagens textuais não são tão intuitivas, e têm uma curva de aprendizagem relativamente grande quando comparadas com linguagens de programação visuais (TAMILSELVAM *et al.*, 2019). Por outro lado, estudos indicam

que a experiência com programação usando ambientes visuais pode ajudar os iniciantes a aprender melhor linguagens tradicionais baseadas em texto uma vez que esses ambientes favorecem o reconhecimento em vez da lembrança além de outras vantagens (BAU et al., 2017).

Desta maneira para introduzir o ensino de ML no ensino médio uma alternativa pode ser criar uma interface visual para um ambiente de desenvolvimento de ML convencional como Jupyter Notebook facilitando em seguida a transição para uma interface textual do mesmo ambiente.

1.2. Objetivos

Objetivo geral

O objetivo geral deste trabalho consiste em desenvolver uma ferramenta visual para o ensino de *Machine Learning* no ensino médio voltada a classificação de imagens. A ferramenta deve fornecer suporte para um *human-centric interactive ML process* (GRESSE VON WANGENHEIM; VON WANGENHEIM, 2021) abordando a análise de requisitos, preparação de dados, seleção e treinamento de modelo de rede neural, configuração dos hiperparâmetros e treinamento, avaliação de desempenho até a exportação do modelo para implantação. A ferramenta deve ser integrada a um *Jupyter Notebook* e executável via *Google Colab*.

Objetivos Específicos

O1. Síntese da fundamentação teórica em relação ao conceito de ensino de *Machine Learning* no ensino médio e ferramentas de suporte para este tema, como funciona, quais tipos de *frameworks* existem atualmente no mercado e ambientes de programação.

O2. Análise do estado da arte em relação a ferramentas para visualizar o processo de desenvolvimento de sistemas de *Machine Learning* para o ensino médio.

O3. Desenvolvimento da biblioteca visual para suportar o processo de desenvolvimento de sistemas de *Machine Learning* para o ensino médio.

O4. Avaliação da qualidade da ferramenta desenvolvida por meio de um painel de especialistas.

1.3. Metodologia de Pesquisa

Nessa pesquisa é usada uma abordagem multi-método. A metodologia de pesquisa utilizada neste trabalho é dividida nas seguintes etapas:

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 as seguintes atividades:

A1.1 - Análise sobre ensino de *Machine Learning* no ensino médio

A1.2 - Síntese dos conceitos de programação visual.

A1.3 - Síntese dos conceitos sobre *Machine Learning*.

A1.4 - Síntese dos conceitos sobre *Jupyter Notebooks* e *Google Colab*.

Etapa 2 – Estado da arte

Nesta etapa é realizado um mapeamento sistemático da literatura seguindo o processo proposto por Petersen et al. (2015) para identificar e analisar como ensinar

ML no ensino médio, ML em geral e ferramentas de programação visuais que atualmente estão sendo utilizadas. Esta etapa é dividida nas seguintes atividades:

A2.1 – Definição do protocolo de revisão

A2.2 – Execução da busca e seleção de artigos relevantes;

A2.3 – Extração e análise de informações relevantes.

Etapa 3 – Desenvolvimento

Nesta etapa será desenvolvido uma ferramenta web para programação visual onde será possível desenvolver modelos de *machine learning* para ser utilizado no contexto de ensinar *machine learning* no ensino médio.

A3.1 - Análise de requisitos.

A3.2 - Modelagem da arquitetura do sistema.

A3.3 - Modelagem de baixo nível e implementação.

A3.4 - Testes do sistema.

Etapa 4 – Avaliação da ferramenta desenvolvida

Nesta etapa o ambiente desenvolvido deverá ser avaliado por meio de painel de especialistas aplicando a ferramenta em disciplina de graduação abordando ML.

A4.1 – Definir a avaliação

A4.2 – Executar a aplicação da ferramenta e coleta de dados.

A4.3 – Analisar os dados coletados.

1.4. Estrutura deste Documento

Na seção 2 deste trabalho é apresentada a fundamentação teórica dos conceitos necessários que suportam a proposta deste trabalho. Na seção 3 é

levantado o estado da arte e a situação atual em que se encontram os trabalhos e propostas existentes sobre o ensino de ML no contexto do ensino médio. Na seção 4 apresenta o desenvolvimento da ferramenta visual e também a aplicação da ferramenta para a tarefa de classificação de imagens. Na seção 5 é apresentado o resultado da avaliação do painel de especialistas. A seção 6 conclui este trabalho, verificando se foram atendidos os objetivos propostos bem como qual a contribuição deste desenvolvimento para a sociedade e também propostas de trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. *Machine Learning*

Machine Learning é uma das subáreas de IA e pode ser caracterizado como o treinamento de um modelo a partir de dados que generaliza uma decisão em relação a uma medida de desempenho (MITCHELL, 1997). Entre as principais tarefas típicas de ML estão o reconhecimento e segmentação de imagens, sendo aplicado em diversos domínios como carros autônomos, saúde, etc. Por exemplo, atualmente já existem clínicas que estão começando a utilizar a detecção e segmentação de objetos em imagens para casos urgentes e sem rápida identificação, como sinalização de oclusão de grandes artérias no cérebro usando imagens radiológicas (BARREIRA et al., 2018). ML também vem sendo cada dia mais utilizado para mecanismos de recomendações, seja em serviços de *streaming* ou sites de compras.

Os algoritmos de ML são capazes de trabalhar com diferentes tipos de dados (imagens, séries temporais, etc.) individualmente, e também podem ser utilizados em conjunto, sendo utilizados em redes neurais de grande escala (ESTEVA et. al., 2019) como por exemplo na Figura 1.

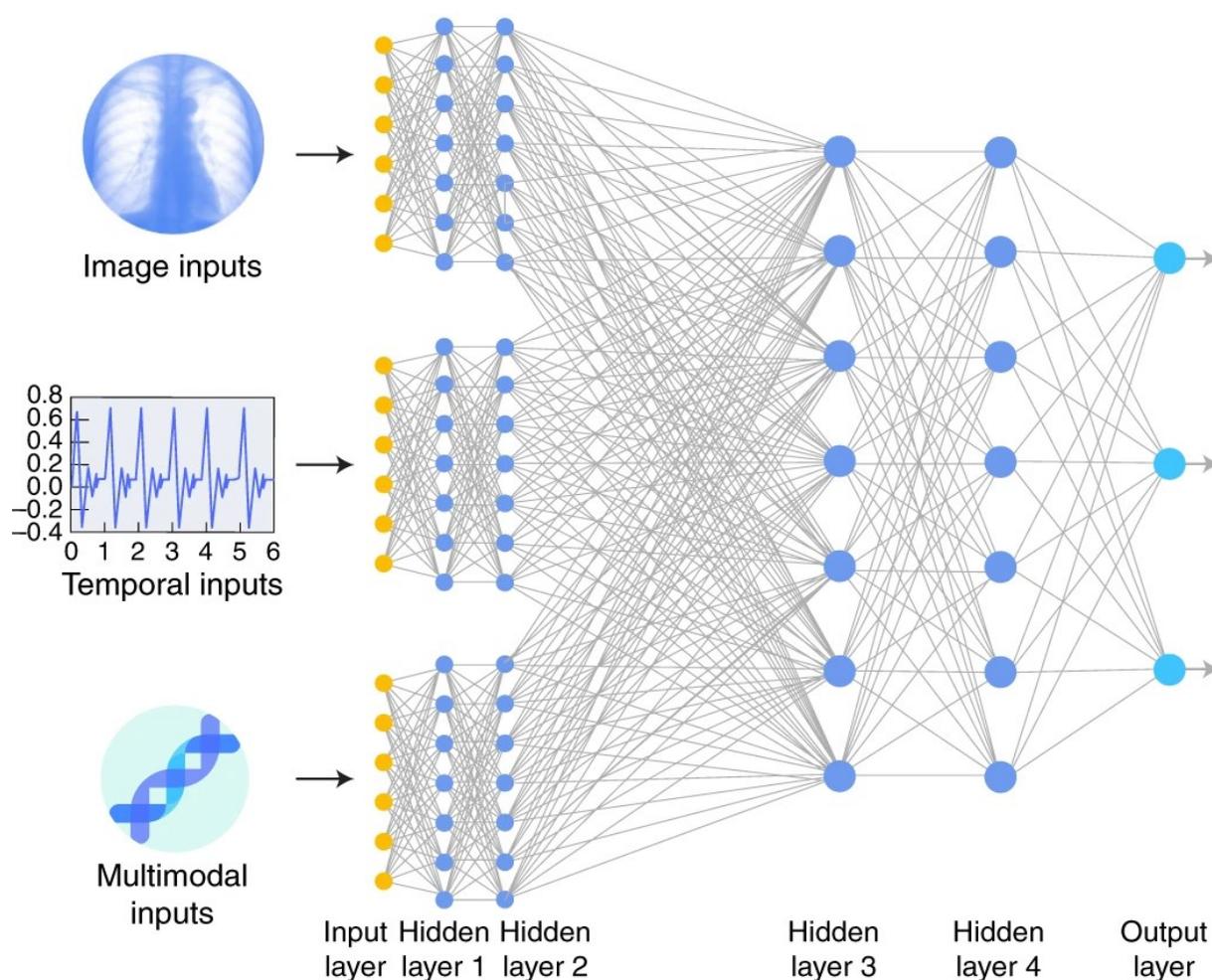


Figura 1 – Exemplo de algoritmo de ML com diferentes tipos de dados de entrada (ESTEVA et al., 2019).

É possível classificar os algoritmos de ML em diversas categorias de acordo com seu estilo de aprendizagem (GOODFELLOW et al., 2016). Na aprendizagem supervisionada, o algoritmo constrói um modelo matemático a partir de um conjunto de dados que contém as entradas e as saídas desejadas. Algoritmos de classificação e algoritmos de regressão são tipos de aprendizagem supervisionada. Na aprendizagem semi supervisionada, uma combinação de dados rotulados e dados não rotulados é usada para fazer melhores previsões para novos pontos de dados do que usando apenas os dados rotulados. No aprendizado não supervisionado, o algoritmo constrói um modelo matemático a partir de um conjunto de dados que contém apenas entradas e nenhum rótulo de saída desejado. Algoritmos de aprendizagem não

supervisionados são usados para encontrar estruturas/padrões nos dados, como agrupar ou agrupar os pontos de dados em categorias. Algoritmos de aprendizagem de reforço recebem *feedback* na forma de reforço positivo ou negativo em um ambiente dinâmico e são usados, por exemplo, em veículos autônomos.

Existem várias bibliotecas que viabilizam o desenvolvimento de modelos de ML com poucas linhas de códigos, alguns exemplos são o *Pytorch* em que é possível rodar o algoritmo com até 75 linhas de código (LUCK, 2012) e também o *Fast.ai* que na sua nova versão é possível construir usando aproximadamente 5 linhas de códigos um classificador de imagem, modelo de segmentação de imagem, sistema de recomendação e etc. (FAST.AI, 2020). Além disso, existem *frameworks* que oferecem uma série de modelos pré-treinados, onde com algumas linhas de código é possível rodar o algoritmo de ML. Um exemplo muito utilizado é o ModelZoo, que faz a curadoria e fornece uma plataforma para pesquisadores de *Deep Learning* encontrarem facilmente modelos pré-treinados para uma variedade de plataformas e usos (MODELZOO, 2019).

A construção de sistemas de ML é um processo iterativo que envolve uma sequência de etapas, que tipicamente incluem (AMAZON, 2019)(GRESSE VON WANGENHEIM; VON WANGENHEIM, 2021):

1. Análise de requisitos. Durante este estágio, o objetivo principal do modelo e seus recursos de destino são especificados. Isso também inclui a caracterização das entradas e saídas esperadas, especificando o problema.
2. Gerenciamento de dados. Durante a coleta de dados, os conjuntos de dados disponíveis são identificados e/ou os dados são coletados. Isso também pode incluir a seleção de conjuntos de dados genéricos disponíveis (por exemplo, ImageNet para detecção de objeto), bem como conjuntos de dados

especializados para aprendizagem por transferência. O tipo de dados depende da tarefa de aprendizado de máquina (por exemplo, imagens, som, texto etc.). Eles também variam muito em termos do número de instâncias, variando de algumas centenas a mais de um bilhão de instâncias. Os dados são preparados pela validação e limpeza dos dados e também podem ser reprocessados transformando os dados brutos. Os conjuntos de dados podem ser rotulados na aprendizagem supervisionada, aumentando cada parte dos dados não rotulados com tags significativas atribuídas manualmente pelos usuários. O conjunto de dados é normalmente dividido em um conjunto de treinamento para treinar o modelo, um conjunto de validação para selecionar o melhor candidato de todos os modelos e um conjunto de teste para realizar uma avaliação de desempenho imparcial do modelo escolhido em dados não vistos (RIPLEY, 2008).

3. Modelo de aprendizagem. Em seguida, um modelo é construído ou mais tipicamente escolhido a partir de modelos bem conhecidos que se mostraram eficazes em problemas ou domínios comparáveis (por exemplo (MODELZOO, 2019)) alimentando os recursos/dados para o algoritmo de aprendizagem. A qualidade do(s) modelo(s) é avaliada a fim de compreender como melhorar iterativamente seu desempenho (por exemplo, em termos de alta precisão, menor erro) testando o modelo contra dados nunca vistos anteriormente (THARWAT, 2019). Hiperparâmetros são propriedades que regem o processo de treinamento tendo um grande impacto no desempenho do modelo que será treinado. Alguns dos principais hiperparâmetros são: o número de etapas de aprendizagem, que define o número de vezes que o algoritmo funcionará em todo o conjunto de dados de treinamento; a taxa de aprendizado, que controla

o quanto o modelo deve aprender a cada etapa de aprendizagem, Valores de inicialização, que definem valores chave que serão utilizados ao treinar o modelo; Tamanho do lote (*batch size*), que define o número de exemplos de treinamento usados em cada etapa de treinamento, etc.

4. Avaliação do modelo. A qualidade do modelo é avaliada a fim de testar o modelo, fornecendo uma melhor aproximação de como o modelo irá se comportar no mundo real, por exemplo, analisando a correspondência entre os resultados do modelo e a marcação humana. A avaliação de modelos de ML não é trivial, e também existem muitos métodos que podem ser aplicados para avaliação de modelo, como validação cruzada, kfold, validação com várias métricas como: acurácia (ACC), precisão, recall, F1, (NGUYEN et al., 2019). De acordo com o domínio específico, diferentes métricas podem ser utilizadas para obter o melhor resultado.
5. Implantação do modelo. Durante a fase de produção/implantação, o modelo é implantado em um ambiente de produção para aplicá-lo a novos eventos de entrada em tempo real. A implantação deste modelo poderá ser incluída em ambientes de programação tradicionais, que utilizam linguagens textuais como Python, mas também nos ambientes baseados em blocos para programação como Scratch, Snap! ou App Inventor. Para uma maior interoperabilidade na comunidade de ferramentas de IA e não ficar preso a uma estrutura ou ecossistema de uma ferramenta específica a ONNX (*Open Neural Networks Exchange*) é a primeira etapa para permitir que mais dessas ferramentas funcionem juntas, permitindo que compartilhem modelos (ONNX, 2021). O objetivo é possibilitar que os desenvolvedores usem as combinações certas de ferramentas para seus projetos. Queremos que todos possam levar a IA da

pesquisa à realidade o mais rápido possível, sem atrito artificial das cadeias de ferramentas. (ONNX, 2021)

Existem várias linguagens de programação populares para ML. Entre elas, Python é a linguagem mais popular, seguida por Java, R e C ++ (Tricon Infotech, 2019). Especialmente no contexto da educação básica, linguagens de programação baseadas em blocos são usadas (WEINTROP, 2019). Esses ambientes melhoram a capacidade de aprendizado para iniciantes, favorecendo o reconhecimento em vez da lembrança; reduzindo a carga cognitiva fragmentando os padrões computacionais em blocos; e usando a manipulação direta de blocos para evitar erros e melhorar a compreensão da estrutura do programa (BAU et al., 2017). Para facilitar o desenvolvimento de modelos de ML estão sendo desenvolvidas ferramentas visuais que permitem aos usuários criar programas arrastando e soltando elementos gráficos, incluindo linguagens baseadas em blocos ou em fluxo, entre outras (WEINTROP, 2019)(BURNETT; BAKER, 1994). As vantagens dessas ferramentas visuais levaram a uma ampla adoção em contextos de programação introdutórios em diferentes estágios educacionais ou qualquer *end-user* (BAU et al. 2017). Especialmente na educação básica, as linguagens de programação baseadas em blocos, como Scratch, SNAP!, Blockly e App Inventor, são proeminentes no ensino de algoritmos e programação (WEINTROP, 2019) e vários desses ambientes de programação baseados em blocos também fornecem extensões para o desenvolvimento de soluções de ML, como para App Inventor (MIT, 2019), Scratch (MIT, 2020) ou SNAP! (MIT, 2020b). Seguindo esse sucesso, várias ferramentas visuais estão sendo propostas também para o ensino de *ML*, as quais são tipicamente formadas por um componente de desenvolvimento de modelos de *ML* e um componente de implantação (GARCIA et al. 2020). O componente de desenvolvimento de modelos de *ML* suporta

a preparação de dados, permitindo coleta e rotulagem dos mesmos, a construção de modelos usando os algoritmos de *ML* disponíveis, a avaliação de desempenho do modelo com dados de teste e, por fim, a exportação do modelo para uma plataforma de programação. Já o componente de implantação suporta desenvolver um sistema de software usando o modelo de *ML* criado previamente, permitindo a criação de soluções inteligentes que tornem o ensino da computação envolvente e motivador.

2.2. Ensino do *Machine Learning* no Ensino Médio

O impacto de IA/ML na sociedade é cada dia maior, mas o ensino de IA no ensino médio ainda não está muito bem definido, apesar de alguns países como a China, por exemplo, já determinarem que todos os alunos do ensino médio aprendam sobre IA no ensino médio (JING, 2018). Internacionalmente as diretrizes curriculares de computação existentes como o *CSTA K-12 Computer Science Framework* (CSTA, 2017) até abordam IA, mas apenas brevemente e no nível do ensino médio, e tratando-se de ML em si, o tema não chega a ser mencionado. No Brasil, a Sociedade Brasileira de Computação (SBC, 2020) fornece as diretrizes para o ensino de computação na educação básica. Ao abordar IA, o tema aparece apenas de forma superficial e sem se aprofundar, além de sempre associada a robótica. Também não é incluído ML ou outras sub-áreas de IA nessas diretrizes de currículo da SBC.

Porém, a intenção de ensino de IA/ML na educação básica não é recente, e até já existiram iniciativas que desde a década de 70 fossem voltadas a ensinar IA nas escolas (PAPERT; SOLOMON, 1971)(KAHN, 1977) e também algumas já na década de 90 até envolviam principalmente redes neurais (BEMLEY, 1999). Mas apenas nos últimos anos houve um aumento do ensino de computação na educação básica. A padronização do que os alunos da educação básica devem saber sobre computação

foi apoiada pelo desenvolvimento de várias diretrizes curriculares, como o CSTA K-12 *Computer Science Framework* (CSTA, 2017). Muitas unidades instrucionais, ferramentas de software e recursos foram desenvolvidos para tornar a computação acessível para os alunos da educação básica desde uma hora de exercícios de programação em código (code.org) a cursos, permitindo-lhes aprender os principais conceitos de computação enquanto criam artefatos significativos que têm impacto direto sobre suas vidas e suas comunidades (TISSENBAUM et al., 2019).

Neste contexto, o *AI for K-12 Working Group* (AI4K12) se juntou as iniciativas da *Association for the Advancement of Artificial Intelligence* (AAAI) e também *Computer Science Teachers Association* (CSTA) com o objetivo de desenvolver diretrizes para o ensino de IA aos alunos do ensino fundamental. São definidas “grandes ideais” em IA que todo aluno deve conhecer para estar de acordo com estas diretrizes (TOURETSKY et al., 2019a):

1. Percepção: os computadores percebem o mundo usando sensores. Os alunos devem entender que a percepção da máquina da linguagem falada ou imagens visuais requer amplo conhecimento do domínio.

2. Representação e raciocínio: os agentes mantêm modelos/representações do mundo e os usam para raciocinar. Os alunos devem entender o conceito de representação e entender que os computadores constroem representações usando dados, e essas representações podem ser manipuladas pela aplicação de algoritmos de raciocínio que derivam novas informações do que já é conhecido.

3. Aprendizagem: os computadores podem aprender com os dados. Os alunos devem entender que o *Machine Learning* é um tipo de inferência estatística que encontra padrões nos dados. As diretrizes da Tabela 1 mostram mais detalhes a

respeito dos objetivos de aprendizado dos jovens do ensino médio desta grande ideia (AI4K12, 2020).

Natureza da aprendizagem	Humanos vs. Máquinas	Definir algoritmos de aprendizado supervisionado, não supervisionado e de reforço, e dar exemplos de aprendizado humano que são semelhantes a cada algoritmo.
	Encontrando padrões nos dados	Modelar como o machine learning constrói um raciocínio para classificação ou previsão ajustando os parâmetros do raciocínio (suas representações internas).
	Treinando um modelo	Usar um algoritmo de aprendizagem supervisionado ou não supervisionado para treinar um modelo em dados do mundo real e, em seguida, avaliar os resultados.
	Construir vs. Usar um Raciocinador ou Reasoner	Ilustrar o que acontece durante cada uma das etapas necessárias ao usar o aprendizado de máquina para construir um classificador ou preditor.
	Ajustando representações internas	Descrever como vários tipos de algoritmos de aprendizado de máquina aprendem ajustando suas representações internas.
	Aprendendo com a experiência	Selecionar o tipo apropriado de algoritmo de machine learning (aprendizado supervisionado, não supervisionado ou de reforço) para resolver um problema de raciocínio.
Redes Neurais	Estrutura de uma rede neural	Descrever as seguintes arquiteturas de rede neural e seus usos: <i>feed-forward network</i> , <i>2D convolutional network</i> , <i>recurrent network</i> , <i>generative adversarial network</i> .
	Ajuste de peso	Treinar uma rede neural multicamadas usando o algoritmo de aprendizado de retro propagação e descrever como os pesos dos neurônios e as saídas das unidades ocultas mudam como resultado do aprendizado....

Conjuntos de dados	Conjuntos de recursos	Comparar dois conjuntos de dados do mundo real em termos dos recursos que eles abrangem e como esses recursos são codificados.
	Grandes conjuntos de dados	Avaliar um conjunto de dados usado para treinar um sistema de IA real, considerando o tamanho do conjunto de dados, a maneira como os dados foram adquiridos e rotulados, o armazenamento necessário e o tempo estimado para produzir o conjunto de dados.
	Polarização	Investigue desequilíbrios nos dados de treinamento em termos de gênero, idade, etnia ou outras variáveis demográficas que possam resultar em um modelo enviesado, usando uma ferramenta de visualização de dados.

Tabela 1 - Objetivos de aprendizado para o ensino médio.

4. Interação natural: fazer os agentes interagirem naturalmente com os humanos é um grande desafio para os desenvolvedores de IA. Os alunos devem entender que, embora os computadores possam compreender a linguagem natural de forma limitada, no momento eles não têm o raciocínio geral e as capacidades de conversação até mesmo de uma criança.

5. Impacto social: as aplicações de IA podem impactar a sociedade de maneiras positivas e negativas. Os alunos devem ser capazes de identificar as formas como a IA está contribuindo para suas vidas, bem como se a construção ética dos sistemas de IA requer atenção às questões de transparência e justiça.

Assim, observa-se que como uma das 5 grandes ideias destas diretrizes também sugerem o ensino de ML na educação básica, inclusive no ensino médio.

Enquanto a IA é "a ciência e a engenharia de fazer máquinas inteligentes que têm a capacidade de atingir objetivos como os humanos", o ML é um subcampo da IA

que lida com o campo de estudo que dá aos computadores a capacidade de aprender sem ser explicitamente programado (Mitchell, 1997). Os algoritmos de ML constroem um modelo matemático baseado em dados de amostra, conhecidos como "dados de treinamento", a fim de fazer previsões ou decisões sem serem explicitamente programados para realizar a tarefa. De acordo com AI4K12, os conceitos de aprendizado de máquina a serem cobertos na educação K-12 devem incluir (TOURETZKY et al., 2019c):

- O que é aprender?
- Abordagens para ML (por exemplo, algoritmos de regressão, algoritmos baseados em instância, máquinas de vetor de suporte, algoritmos de árvore de decisão, algoritmos Bayesianos, algoritmos de agrupamento, algoritmos de rede neural artificial).
- Tipos de algoritmos de aprendizagem por estilo de aprendizagem.
- Fundamentos de redes neurais.
- Tipos de arquitetura de rede neural.
- Como os dados de treinamento influenciam o aprendizado.
- Limitações do aprendizado de máquina.

Para ser efetivo, um programa de ensino deve incluir desde a compreensão de conceitos básicos de *ML* até competências relacionadas à aplicação, ensinando os estudantes a criar modelos de *ML* na forma de “aprender fazendo”. Desse modo, ao vivenciarem as possibilidades, pontos fortes e fracos dessa tecnologia, acredita-se que possam alcançar uma compreensão mais profunda do tema (WONG et al., 2020)(MARQUES et al., 2020).

O desenvolvimento de aplicativos de *ML* não é trivial e o processo de desenvolvimento difere do de um software tradicional, pois envolve adquirir um

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

2.3. Ambientes de programação visuais

Dada a sua complexidade, os modelos de *ML* normalmente são desenvolvidos usando linguagens de programação baseadas em texto que requerem codificação, envolvendo uma compreensão dos conceitos de programação e sua sintaxe, aumentando significativamente a carga cognitiva e a curva de aprendizagem (MCCRACKEN et al., 2001). Portanto, para popularizar *ML*, é desejável reduzir o esforço cognitivo para que o usuário possa se concentrar na lógica para resolver o problema em questão (KNUTH; PARDO, 1980). Com esta finalidade, foram introduzidas linguagens visuais que permitem aos usuários criar programas simplesmente arrastando e soltando um elemento visual em uma tela e, posteriormente, conectando esse elemento a outros elementos, em vez de especificá-los textualmente (IDRESS et al., 2018)(WEINTROP; WILENSKY, 2017). Essas

representações visuais podem assumir diversas formas, incluindo linguagens baseadas em blocos ou em fluxo (BURNETT; BAKER, 1994)(PASTERNAK et al., 2017). As linguagens visuais podem melhorar a capacidade de aprendizagem de novatos, ajudando-os a evitar erros, favorecer o reconhecimento em vez de recordar e fornecer conjuntos de instruções limitados específicos de domínio, reduzindo a carga cognitiva (ÇAKIROĞLU et al., 2018). Essas vantagens levaram à ampla adoção em contextos introdutórios de programação em diferentes estágios educacionais (BAU et al., 2017). Atualmente as novas gerações de ambientes baseados em blocos vêm sendo liderada pela popularidade do Scratch (RESNICK et al. 2009), Snap! (HARVEY; MÖNIG 2010), Blockly (FRASER 2015) entre outros.

Seguindo esse sucesso, ferramentas visuais também estão sendo propostas para o ensino de ML. Essas ferramentas normalmente incluem um componente para o desenvolvimento de um modelo de ML e um componente de implantação (RODRÍGUEZ-GARCÍA et al., 2020). Para o desenvolvimento de aplicações de IA com ferramentas de programação baseadas em blocos é possível identificar dois tipos de ferramentas de suporte para o desenvolvimento de modelos de ML: sendo o primeiro os ambientes de programação baseados em blocos em si, como por exemplo o *eCraft2Learn* (KAHN; WINTERS. 2018), que permite desenvolver o modelo de ML encaixando blocos como um quebra-cabeça para montar a solução desejada. Já o segundo são os ambientes de programação baseadas em fluxo, que são linguagens visuais em que blocos (também chamados de nós) são conectados por arcos (ou fios). Portanto, um programa em uma linguagem de programação baseada em fluxo é um gráfico direcionado através do qual os dados fluem entre os blocos e cada bloco fornece uma função que pode transformar os dados recebidos (JOHNSTON et al., 2004)(HILS, 1992).

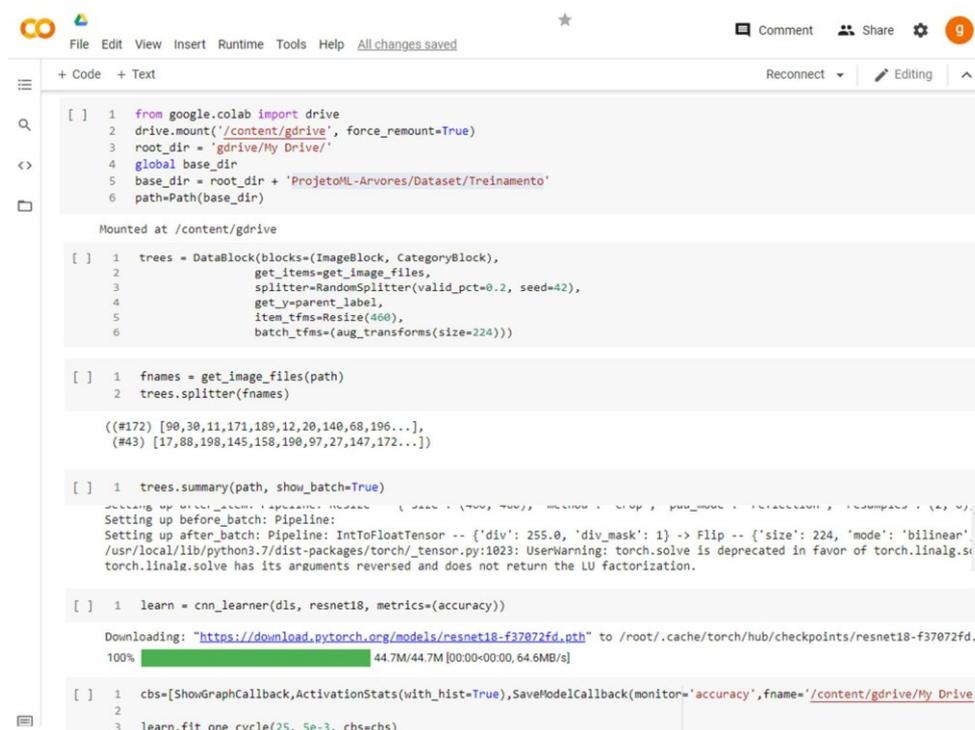
Estes ambientes baseados em fluxo por sua vez podem ser divididos em dois tipos: Ambientes de programação baseados em fluxos de trabalho, que normalmente possuem o fluxo de trabalho em si bem definido, e apresentam dentro dos blocos as etapas necessárias como por exemplo o Google Teachable Machine. E também os ambientes de programação baseados em fluxos de dados, em que é possível montar da forma que se desejar todo o processo idealizado, não ficando necessariamente preso às etapas definidas pelo criador da ferramenta.

2.4. Jupyter Notebook e Google Colab

Um dos ambientes convencionais mais populares usados atualmente para desenvolver modelos de ML é o Notebook Jupyter (PERKEL, 2018). O Jupyter Notebook¹ é um aplicativo gratuito de código aberto que permite aos usuários criar e compartilhar documentos que contêm código de software executável, saída computacional, texto explicativo e recursos multimídia em um único documento. Também permite a combinação de diversas linguagens de programação, como Python, R, etc. Os notebooks Jupyter são normalmente usados como ambientes de programação baseados em texto para desenvolver modelos de ML usando, por exemplo, Python. É um ambiente de computação interativo, acessível por meio de qualquer navegador da web, que permite aos usuários usar, modificar ou criar documentos da web multimídia interativos. Os documentos do Jupyter Notebook fornecem um registro completo e executável de computação que pode ser compartilhado com outras pessoas e fornece um registro completo e executável do processo. Os arquivos de notebook criados são um formato JSON simples e

¹ jupyter.org

documentado, com a extensão '.ipynb', que pode ser facilmente acessado e/ou manipulado por outras ferramentas de software.



```

File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text Reconnect Editing
[ ] 1 from google.colab import drive
2 drive.mount('/content/gdrive', force_remount=True)
3 root_dir = 'gdrive/My Drive/'
4 global base_dir
5 base_dir = root_dir + 'ProjetoML-Arvores/Dataset/Treinamento'
6 path=Path(base_dir)

Mounted at /content/gdrive

[ ] 1 trees = DataBlock(blocks=(ImageBlock, CategoryBlock),
2 get_items=get_image_files,
3 splitter=RandomSplitter(valid_pct=0.2, seed=42),
4 get_y=parent_label,
5 item_tfms=Resize(460),
6 batch_tfms=(aug_transforms(size=224)))

[ ] 1 fnames = get_image_files(path)
2 trees.splitter(fnames)

((#172) [90,30,11,171,189,12,20,140,68,196...],
(#43) [17,88,198,145,158,190,97,27,147,172...])

[ ] 1 trees.summary(path, show_batch=True)
Setting up before_batch: Pipeline:
Setting up after_batch: Pipeline: IntToFloatTensor -- {'div': 255.0, 'div_mask': 1} -> Flip -- {'size': 224, 'mode': 'bilinear'}
/usr/local/lib/python3.7/dist-packages/torch/_tensor.py:1023: UserWarning: torch.solve is deprecated in favor of torch.linalg.solve
torch.linalg.solve has its arguments reversed and does not return the LU factorization.

[ ] 1 learn = cnn_learner(dls, resnet18, metrics=(accuracy))

Downloading: "https://download.pytorch.org/models/resnet18-f37072fd.pth" to /root/.cache/torch/hub/checkpoints/resnet18-f37072fd.
100% 44.7M/44.7M [00:00<00:00, 64.6MB/s]

[ ] 1 cbs=[ShowGraphCallback,ActivationStats(with_hist=True),SaveModelCallback(monitor="accuracy",fname="/content/gdrive/My Drive
2
3 learn.fit_one_cycle(25, 5e-3, cbs=cbs)

```

Figura 2. Exemplo de uma interface textual de um Jupyter Notebook no Google Colab.

É também um excelente ambiente de aprendizagem para os estudantes (BARBA et al., 2019), oferecendo uma ferramenta valiosa para o ensino em direção à compreensão, movendo os alunos, por exemplo, da visualização passiva do conteúdo do curso para explorar, analisar, sintetizar e criar conteúdo de maneiras ativas. Para começar, os alunos podem consumir o conteúdo do Notebook lendo e executando Notebooks no estágio de "uso" do ciclo de usar-modificar-criar (LYTLE et al., 2019) e, em seguida, passar para editar ou completar cadernos no estágio de "modificação", além de criar suas próprias soluções no estágio de "criação". Incentivar hábitos de aprendizagem ativa, por meio do uso de Notebooks interativos, também proporcionará benefícios que impulsionam o engajamento, o interesse e a exploração de conceitos (BARBA et al., 2019) e aumentam o desempenho (FREEMAN et al., 2014). Além

disso, os Notebooks Jupyter são fáceis de usar, dispensando instalação e podem ser adotados em diversos tipos de cursos (online ou presencial).

O Jupyter notebook possui dois componentes. Os usuários inserem o código de programação ou texto em células retangulares em uma página da web de front-end. O navegador então passa esse código para um *back-end* 'kernel', que executa o código e retorna os resultados. É importante ressaltar que os *kernels* não precisam residir no computador do usuário, pois os notebooks também podem ser executados na nuvem. Por exemplo, o *Colaboratory do Google* (abreviadamente *Google Colab*)² fornece um front-end com o tema do Google para o bloco de notas Jupyter. Ele permite que os usuários colaborem e executem códigos inteiramente na nuvem, explorando os recursos de nuvem do Google, como unidades de processamento gráfico, e salvem facilmente seus artefatos no Google Drive. Isso permite treinar modelos de ML mesmo sem acesso a uma máquina potente ou acesso à Internet de alta velocidade. O *Google Colab* oferece suporte a instâncias de *GPU* e *TPU*, o que o torna uma ferramenta perfeita para aprendizado profundo. Como um *notebook Colab* pode ser acessado remotamente de qualquer máquina por meio de um navegador que não requer configuração para uso, ele também é adequado para fins educacionais. O Colab também permite que os usuários usem e compartilhem notebooks Jupyter com outras pessoas sem precisar baixar, instalar ou executar nada, pois os notebooks Colab são armazenados no Google Drive ou podem ser carregados do GitHub. Além disso, também permite fazer *upload* de artefatos, como imagens diretamente de um Google Drive. Isso o torna prático para uso também em contexto educacional, no qual, por exemplo, um professor pode configurar um notebook e dar acesso facilmente aos alunos ou alunos trabalhando juntos em um projeto (KLUYVER et al., 2016).

² <https://colab.research.google.com>

Além da forma comum de programação baseada em texto nos Notebooks Jupyter, a extensão IPyWidgets³ fornece *widgets* HTML interativos para notebooks Jupyter e o *kernel* IPython, o que permite uma interface semelhante à GUI interação gráfica com notebooks Jupyter por meio de menus de seleção, barras deslizantes, botões de rádio/alternância ou caixas de texto. Esses *widgets* gráficos podem ser incorporados aos Jupyter notebooks e os usuários podem combinar a análise de script usual com a ativação de tais *widgets* quando desejado. Eles podem ser usados, por exemplo, para variar os valores dos parâmetros de entrada e explorar um conjunto de dados ou resultados computacionais. Esses elementos podem ocultar o código baseado em texto e permitir a criação de um bloco de notas com uma interface visual. Atualmente, IPyWidgets são usados principalmente para análise e visualização de dados, permitindo transformar Jupyter Notebooks em painéis interativos para explorar e visualizar dados (PERKEL, 2018)(PIAZENTIN ONO et al., 2021).

³ <https://ipywidgets.readthedocs.io/en/>

3. ESTADO DA ARTE

Para obter o estado da arte e prática sobre se e como ML educacional é dirigida no ensino médio, foi realizado um estudo de mapeamento sistemático seguindo o procedimento proposto por Petersen et al. (2015).

3.1. Definição do protocolo de revisão

O objetivo deste estudo é responder à pergunta de pesquisa: Quais ferramentas visuais existem para o ensino de ML no médio por meio do desenvolvimento de modelos de ML personalizados? O objetivo deste trabalho é caracterizar e comparar essas ferramentas, para elaborar uma ferramenta capaz de atender as necessidades. Portanto, analisamos as seguintes questões:

PA1. Quais ferramentas visuais existem para ensinar ML?

PA2. Quais são suas características educacionais?

PA3. Quais são suas características em relação à plataforma de ML?

PA4. Quais são suas características em relação à plataforma de implantação?

PA5. Como as ferramentas foram desenvolvidas e avaliadas?

Critérios de inclusão e exclusão: Foram consideradas apenas publicações em inglês desde 2010 que apresentam uma ferramenta visual para o desenvolvimento de modelos de ML, não incluindo linguagens de programação visual genéricas ou ferramentas para outros domínios. Os esforços foram concentrados em ferramentas que permitem criar modelos de ML personalizados, excluindo ferramentas para fins de demonstração. Também excluímos qualquer abordagem com foco apenas na visualização de modelos de ML ou visando a automação completa do seu

desenvolvimento. Além disso, foram incluídas apenas ferramentas que foram desenvolvidas ou usadas para fins educacionais no ensino médio. Conseqüentemente, foram excluídas qualquer ferramenta de ML direcionada exclusivamente para usuários finais profissionais ou adultos. Consideramos apenas os artigos que apresentam informações substanciais permitindo a extração de informações relevantes sobre as questões de análise. Portanto, artigos de apenas um resumo ou de uma página são excluídos.

Fontes: Foram pesquisados os principais bancos de dados e bibliotecas digitais da área de computação, incluindo *ACM Digital Library*, *IEEE Xplore Digital Library* e *Scopus* com acesso via Portal da Capes. Também pesquisamos no Google para encontrar ferramentas que não foram publicadas em bibliotecas científicas. Para minimizar ainda mais o risco de omissão, também incluímos a literatura encontrada por meio da técnica de *backward* e *forward snowballing* (WOHLIN, 2014). A literatura secundária foi consultada para completar as informações sobre as ferramentas encontradas.

Definição da *string* de pesquisa. Com base na pergunta de pesquisa, várias pesquisas informais foram realizadas para calibrar a string de pesquisa, identificando os termos de pesquisa relevantes (Tabela 2). Também incluímos sinônimos para minimizar o risco de omissão de obras relevantes. Não incluímos termos relacionados à educação, pois isso em pesquisas de teste retornou principalmente artigos relacionados à aplicação de técnicas de ML para análise de aprendizagem ou aprendizagem personalizada, em vez de estar relacionado ao ensino de conceitos de ML.

Termo	Sinônimos	Tradução para inglês
Machine Learning	Rede neural	<i>Machine learning, neural network</i>
Linguagem de programação em blocos	Linguagens de programação visuais, <i>gui tool, toolkit</i>	<i>Block-based languages, Visual programming language, gui tool, toolkit</i>

String de busca genérica

((("visual programming" OR "block-based programming" OR "gui tool" OR "toolkit") AND ("machine learning" OR "neural network"))) AND Publication Date 2010 TO 2020

Fonte	String de pesquisa
ACM Digital Library	[Abstract: "visual programming"] OR [Abstract: "block-based programming"] OR [Abstract: "gui tool"] OR [Abstract: toolkit] AND [[Abstract: "machine learning"] OR [Abstract: "neural network"]] AND [Publication Date: (01/01/2010 TO 12/31/2020)]
IEEE Xplore Digital Library	((("Abstract": "visual programming" OR "Abstract": "block-based programming" OR "Abstract": "gui tool" OR "Abstract": "toolkit") AND ("Abstract": "machine learning" OR "Abstract": "neural network")))) Filters Applied: 2010 - 2020
Scopus	TITLE-ABS-KEY (("visual programming" OR "block-based programming" OR "gui tool" OR toolkit) AND ("machine learning" OR "neural network")) AND PUBYEAR > 2010 AND (LIMIT-TO (SUBJAREA, "COMP"))
Google	"block-based" "machine learning"

Tabela 2: Termos de Busca e String de pesquisa para cada fonte.

3.2. Execução da busca

A pesquisa foi realizada em novembro de 2020 pelo autor atualizando e customizando a revisão da literatura feita por Gresse von Wangenheim *et al.* (2020). A busca inicial recuperou um total de 1203 artigos nas bases científicas e 780.000 artigos no Google (Tabela 3). Devido ao grande número de resultados de algumas pesquisas, restringimos a análise às 300 mais relevantes. Analisamos rapidamente os resultados da pesquisa com base em seu título e resumo. Publicações irrelevantes e duplicadas retornadas por múltiplas pesquisas foram removidos. Esse estágio resultou em 25 artigos potencialmente relevantes. Na segunda etapa de seleção, analisamos o texto completo aplicando os critérios de inclusão e exclusão para identificar os relevantes.

Fonte	Quantidade de artigos resultantes da busca	Quantidade de artigos analisados	Quantidade de artigos potencialmente relevantes (título e abstract)	Quantidade de artigos relevantes (com base no artigo na íntegra)
ACM	254	254	4	2
IEEE	296	296	3	2
SCOPUS	653	300	5	4
Google	780000	300	4	0
Backward snowballing			5	3
Forward snowballing			4	0
Total (sem duplicados)				11

Tabela 3 – Número de artigos identificados por estágio de seleção

Focando especificamente em ML, excluimos qualquer ferramenta que fornecesse suporte geral para o ensino de IA, como o Logo (KAHN, 1977). Visando o suporte ao desenvolvimento de modelos de ML customizados, também excluimos ferramentas apenas para demonstração e visualização, como Tensorflow Playground (SMILKOV et al., 2017), TensorBoard (WONGSUPHASAWAT et al., 2018) e DeepGraph (HU et al., 2018). Também excluimos ambientes que suportam exercícios bem definidos e controlados, como a IA do code.org para atividade dos oceanos, Zhorai (LIN et al., 2020), PopBots (WILLIAMS et al., 2019) ou IA conversacional (VAN BRUMMELEN et al., 2019), orientando o desenvolvimento de um modelo específico (ou partes) e/ou apenas sua implantação. Também excluimos ferramentas visuais voltadas para o uso profissional e para as quais não foram relatadas aplicações na educação K-12, como Apple Machine Learning, KNIME, Microsoft Azure, Nvidia Digits, Sony Neural Network Console, (HAUCK et al., 2019), (XIE et al., 2019), entre outros.

Foram excluídos desta pesquisa quaisquer ferramentas voltadas a outros níveis educacionais (ensino infantil, fundamental e superior) ou que não especificasse o ensino médio (ZIMMERMANN-NIEFIELD et al., 2019b)(QUEIROZ et al., 2018)(DRUGA, 2018)(DRUGA et al., 2019)(KAHN; WINTERS, 2017)(CARNEY et al.,

2020)(RODRÍGUEZ-GARCÍA et al., 2020)(LANE, 2018)(DEMŠAR et al., 2013)(GODEC et al., 2019)(SAKULKUEAKULSUK et al., 2018)(AGASSI et al., 2019). Também excluímos extensões baseadas em bloco, como BlockPy (BART et al., 2017) ou Jigsaw, fornecendo uma combinação de interfaces baseadas em bloco com Python em notebooks Jupyter, visto que são voltadas para estudantes de ensino superior. Aplicando *Backward* e *Forward snowballing* com base nos estudos primários, foram identificados 3 artigos adicionais. Como resultado, um total de 11 artigos relevantes foram identificados, representando 7 ferramentas.

3.3. Análise dos resultados

3.3.1. Quais ferramentas visuais existem para ensinar ML no ensino médio?

Foram identificadas 7 ferramentas visuais de programação desenvolvidas ou sendo usadas para ensinar o processo de criação de modelos personalizados de ML no ensino médio (Tabela 4).

Nome	Descrição breve	Escopo		Referência(s)
		Plataforma de ML	Plataforma para deploy	
AlpacaML	Um aplicativo iOS que oferece suporte aos usuários na criação, teste, avaliação e uso de modelos de ML de gestos com base em dados de sensores vestíveis.	X	Scratch	<p>A. Zimmermann-Niefield, S. Polson, C. Moreno, and R. B. Shapiro. 2020. Youth making machine learning models for gesture-controlled interactive media. In Proceedings of the Interaction Design and Children Conference. ACM, 63–74.</p> <p>A. Zimmermann-Niefield, M. Turner, B. Murphy, S. K. Kane, and R. B. Shapiro. 2019b. Youth Learning Machine Learning through Building Models of Athletic Moves. In Proceedings of the 18th International Conference on Interaction Design and Children. ACM, 121–132.</p>

eCraft2learn	Blocos adicionais para a linguagem de programação visual Snap! que fornece interface fácil de usar para os serviços de nuvem de IA e funcionalidade de aprendizado profundo.	X	Snap!	K. M. Kahn and N. Winters. 2018. AI Programming by Children. In Proceedings of the Conference on Constructionism, Vilnius, Lithuania.
				K. M. Kahn, R. Megasari, E. Piantari, and E. Junaeti. 2018. AI Programming by Children Using Snap! Block Programming in a Developing Country. In Proceedings of the 13th European Conference on Technology Enhanced Learning, Leeds, UK, 2018.
				K. M. Kahn, Y. Lu, J. Zhang, N. Winters, and M. Gao. 2020. Deep learning programming by all. In Proceedings of the Conference on Constructionism, Dublin, Ireland.
<i>Educational Approach to ML with Mobile Applications</i>	Um conjunto de extensões do App Inventor abrangendo vários subcampos de ML, entre os quais a extensão Teachable Machine permite desenvolver um modelo de ML.	X	App Inventor	K. Zhu. 2019. An Educational Approach to Machine Learning with Mobile Applications. M.Eng thesis, MIT, Cambridge, MA, USA.
Milo	Um ambiente de programação visual baseado na web para Educação em Ciência de Dados.	X	-	A. Rao, A. Bihani, and M. Nair. 2018. Milo: A visual programming environment for Data Science Education. In Proceedings of the Symposium on Visual Languages and Human-Centric Computing, Lisbon, Portugal, 211-215.
PIC (<i>Personal Image Classifier</i>)	Um sistema web onde os usuários podem treinar, testar e analisar modelos de classificação de imagem personalizados com uma extensão para MIT App Inventor que permite usar os modelos em aplicativos.	X	App Inventor	D. Tang, Y. Utsumi, and N. Lao. 2019. PIC: A Personal Image Classification Webtool for High School Students. In Proceedings of the IJCAI EduAI Workshop, Macao, China.
				D. Tang. 2019. Empowering Novices to Understand and Use Machine Learning With Personalized Image Classification Models, Intuitive Analysis Tools, and MIT App Inventor, M.Eng thesis, MIT, Cambridge, USA.
snAlp	Uma estrutura que permite a aprendizagem construcionista da	X	Snap!	S. Jatzlau, T. Michaeli, S. Seegerer and R. Romeike. 2019. It's not Magic After All – Machine Learning in Snap! using Reinforcement Learning. In Proceedings of IEEE Blocks and

	aprendizagem por reforço com SNAP!			Beyond Workshop, Memphis, TN, USA, 37-41.
DeepScratch	Uma ferramenta que permite a extensão do Scratch com blocos de IA/ML	X	Scratch	N. Alturayeif, N. Alturaief, Z.Alhathlou DeepScratch: Scratch Programming Language Extension for Deep Learning Education. International Journal of Advanced Computer Science and Applications(IJACSA), Volume 11 Issue 7, 2020.

Tabela 4 - Ferramentas visuais para ensinar ML no ensino médio

As ferramentas analisadas oferecem suporte parcial e/ou total para o desenvolvimento de modelos de ML bem como sua implantação como parte de artefatos de software, como por exemplo em jogos ou aplicativos móveis. A maioria das ferramentas oferece a implantação integrada a ambientes de programação baseados em blocos comumente utilizados no ensino médio, como Scratch, App Inventor e Snap! (Figura 3).

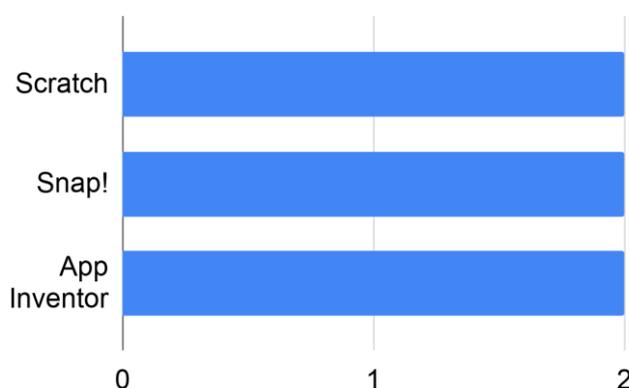


Figura 3 - Ambientes com suporte para a implantação dos modelos de ML criados

A maioria das ferramentas está disponível online gratuitamente, mas alguns requerem o registro do usuário e/ou o uso de chaves API, o que aumenta a complexidade, uma vez que público-alvo não tem domínio em programação em geral, dificultando o uso (Tabela 5). A maioria das ferramentas está disponível apenas em inglês, com exceção do SnAIP que também oferece suporte ao alemão. Nenhuma delas é oferecida em português.

Nome	Site da versão em execução da ferramenta	Plataforma	Licença de uso	Registro de usuário / Chave API	Idioma(s)
AlpacaML	-	online/app	NI	Chave para conectar com Scratch	Inglês
eCraft2learn	https://ecraft2learn.github.io/ai	online	Gratuito	Necessário	Inglês
<i>Educational Approach to ML with Mobile Applications</i>	https://appinventor.mit.edu/explore/resources/ai/image-classification-look-extension	online	Gratuito	-	Inglês
Milo	https://miloide.github.io/	online	Gratuito	-	Inglês
PIC (<i>Personal Image Classifier</i>)	https://classifier.appinventor.mit.edu	online	Gratuito	-	Inglês
SnAlp	https://ddi.cs.fau.de/schule/snaip	online	Gratuito	NI	Inglês, Alemão
DeepScratch	https://github.com/Noufst/DeepScratch	online	-	-	Inglês

Tabela 5: Características gerais das ferramentas.

3.3.2. Quais são suas características educacionais?

Toda a ferramenta selecionada tem seu público alvo como jovens do ensino médio, porém algumas também são destinadas a outros níveis de ensino, bem como não especialistas. Utilizando uma estratégia de aprender fazendo, as ferramentas são utilizadas em unidades extracurriculares de curta duração, como em workshops. Já em outra abordagem, foi adotado como parte de um curso de ML mais abrangente ao longo de seis semanas, cobrindo também, de forma mais ampla, mais conhecimento teórico e/ou discutindo o impacto ético e social do ML, bem como oportunidades de carreira.

Algumas ferramentas são acompanhadas por unidades educacionais que são apenas uma atividade prática normalmente guiada por um tutorial passo a passo e artefatos de exemplo ou, em alguns casos, também incluem uma parte mais teórica na forma de aulas expositivas e/ou vídeos. Vários seguem o ciclo de usar-modificar-

criar, incentivando os alunos também a criar seu modelo de ML personalizado no estágio final da unidade educacional (Tabela 6). Adotando uma estratégia de ação computacional, vários cursos incluem a implantação dos modelos de ML desenvolvidos, permitindo aos alunos criar uma solução inteligente de trabalho na forma de um aplicativo móvel ou jogo. Aplicando conceitos de ML através da criação de modelos de ML, essas unidades também provocam análises críticas dos resultados de desempenho obtidos, bem como dos pontos fortes e fracos do ML em geral. Vários graus de suporte educacional acompanham as ferramentas, incluindo principalmente tutoriais passo a passo para atividades práticas usando as ferramentas. Outros materiais educacionais incluem planos de aula, slides, vídeos, exemplos e exercícios.

Nome	Público Alvo	Unidades educacionais de acompanhamento		
		Tipo e duração	Estratégia Educacional	Recursos Educacionais
AlpacaML	Alunos do ensino fundamental e médio sem experiência em ML	Workshop de 3 horas	A ferramenta é demonstrada e os alunos aprendem a construir um modelo predefinido seguindo um tutorial interativo. Em seguida, os alunos constroem modelos de sua própria atividade física, coletando, rotulando dados e avaliando o modelo.	NI
eCraft2learn	Alunos do ensino básico em geral e não-especialistas	NI	Os alunos primeiro discutem exemplos de aplicativos de IA e são apresentados ao Snap! e como usar seus blocos AI. Em seguida, eles experimentam a síntese de voz e criam programas usando o reconhecimento de imagem.	Processo de aprendizagem, tutoriais e exercícios, programas de exemplo, vídeos
<i>Educational Approach to ML with Mobile Applications</i>	Alunos do ensino médio	Curso de 6 semanas	Um currículo simples usando as extensões de uma série de tutoriais. Cada aula é dividida em duas partes: palestra (até a primeira metade da aula) e o desenvolvimento de aplicativos móveis.	Esboços de aulas, tutoriais interativos e em vídeo, slides

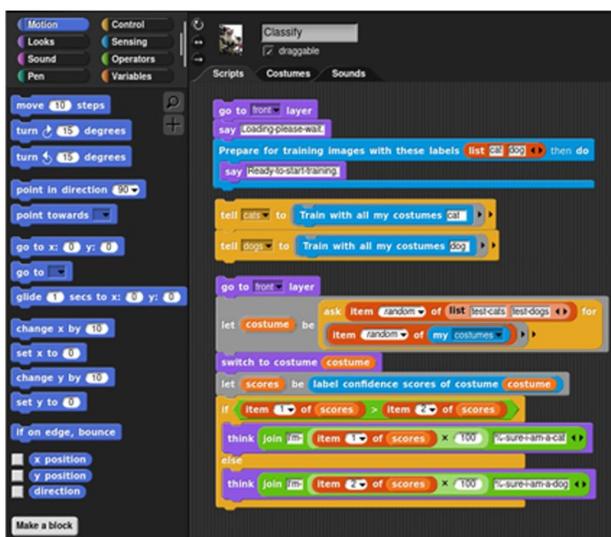
Milo	Alunos do Ensino médio a curso superior sem experiência em programação e que não estudam áreas da computação	--	--	--
PIC (<i>Personal Image Classifier</i>)	Alunos do Ensino médio	Workshop com duas aulas de 50 minutos	As primeiras aulas começam com uma breve introdução aos conceitos básicos de ML, então os alunos usam o PIC para construir um modelo de ML. Na segunda aula, os alunos usam a extensão com os modelos treinados para criar aplicativos móveis inteligentes.	Plano de aula, guia do professor, slides, tutoriais
SnAlp	Alunos do ensino médio	NI	Introdução à aprendizagem por reforço, atividade prática seguindo um tutorial pré-definido incluindo a implantação, avaliação e otimização do modelo.	Tutoriais, exemplos de solução, cartões, instruções do professor (apenas em alemão)
DeepScratch	Alunos do ensino médio	NI	--	--

Tabela 6 - Características educacionais das ferramentas.

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

Foram identificados dois tipos de ferramentas de suporte para o desenvolvimento de modelos de ML (Figura 4). Quatro ferramentas (eCraft2Learn, Milo, SnAlp e DeepScratch) fornecem suporte baseado em bloco ao estender o respectivo ambiente de programação, fornecendo blocos de ML específicos para preparação, treinamento e avaliação de dados. Apenas três ferramentas (AlpacaML, *Educational Approach to ML with Mobile Applications* e PIC) adotam uma abordagem

baseada em *workflow* para apoiar o desenvolvimento do ML, guiando o usuário passo a passo por meio de um navegador da web ou usuário visual baseado em aplicativo.



Exemplo baseado em blocos (eCraft2Learn)

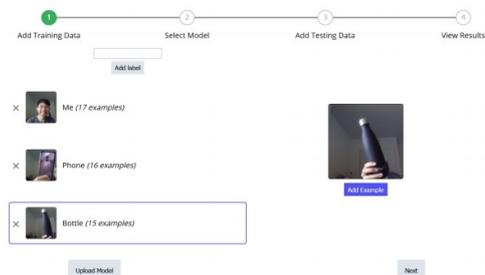


Figure 2: The UI for the training step. Users can easily add labels and examples here.



Exemplo baseado em workflow (PIC)

Figura 4 - Exemplos de suporte ao desenvolvimento de ML

Todas as ferramentas são limitadas quanto à tarefa de ML que suportam, focando principalmente no reconhecimento de imagens, sendo as tarefas em que os aplicativos de ML atuais estão tendo muito sucesso (Figura 5). Apenas o eCraft2learn recentemente também adicionou blocos de suporte à detecção e segmentação de objetos, enquanto Zhu (2019) oferece suporte à detecção de objetos. Outras tarefas cobertas incluem síntese de voz (eCraft2Learn), clustering (Milo) e um agente de jogo (SnAlp).

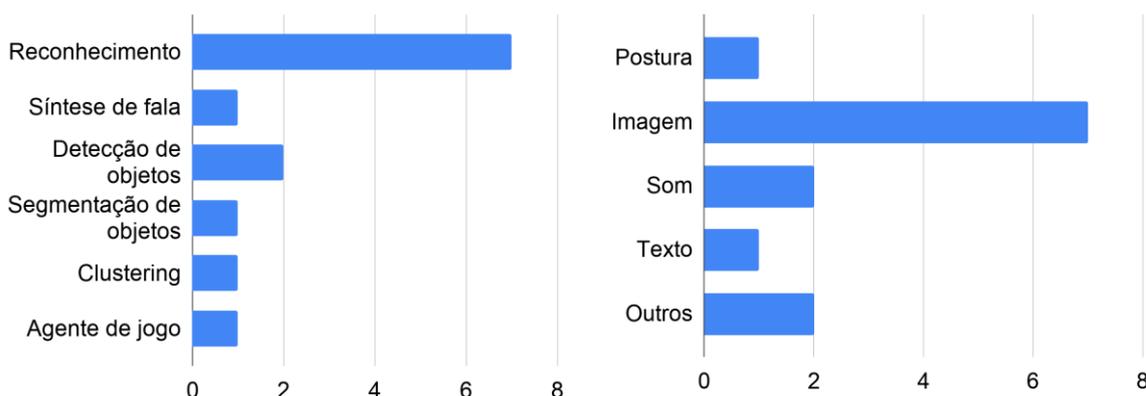


Figura 5 - Frequência de tarefas de ML e tipos de dados compatíveis

As ferramentas suportam amplamente todas as etapas básicas do processo de desenvolvimento de ML (Figura 6): primeiro, elas incentivam a coleta de pequenas quantidades de dados e sua rotulagem, organizando-os em categorias criadas pelo usuário. Em seguida, esses dados são usados para treinar um modelo de ML usando o aprendizado por transferência, o que permite construir modelos precisos de forma que economize tempo, usando back-ends de ML como por exemplo o TensorFlow. Uma vez que o modelo é treinado, seu desempenho pode ser avaliado. Isso é feito principalmente permitindo que o usuário teste o modelo com novos dados para os quais o modelo fornece como saída o rótulo previsto. Apenas uma ferramenta fornece suporte para analisar o desempenho por imagem (PIC). Apenas uma ferramenta oferece suporte à exportação do modelo criado para sua implantação em Python (Milo). Nenhuma das ferramentas oferece suporte à análise de requisitos. A engenharia de recursos também não é oferecida

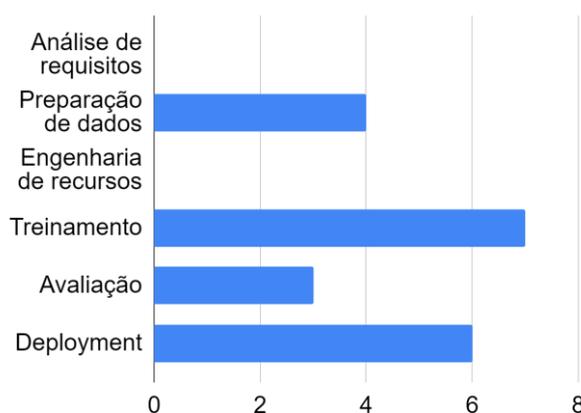


Figura 6 - Frequência das etapas do processo de ML com suporte

De forma geral, as ferramentas suportam uma forma interativa que permite aos alunos fazerem as correções necessárias de forma informada e sistemática. O suporte abrangente para o processo completo de ML, incluindo a oportunidade para os alunos realizarem a preparação e avaliação de dados, também permite que eles construam um entendimento mais preciso.

Nome	Tipo de Ferramenta	Tarefas de ML suportadas	Partes do processo de ML suportadas
AlpacaML	Workflow	Reconhecimento de movimento	Preparação de dados, treinamento, avaliação, implantação
eCraft2learn	Baseada em blocos	Reconhecimento de imagem e fala, síntese de fala, detecção de objetos e segmentação	Preparação de dados, treinamento, avaliação, implantação
<i>Educational Approach to ML with Mobile Applications</i>	Workflow	Reconhecimento de imagem e texto, detecção de objetos	Preparação de dados, treinamento, implantação
Milo	Baseada em blocos	<i>Clustering</i>	Treinamento
PIC (<i>Personal Image Classifier</i>)	Workflow	Reconhecimento de imagem	Preparação de dados, treinamento, avaliação, implantação
SnAlp	Baseada em blocos	Agente de jogo	Training, Deployment
DeepScratch	Baseada em blocos	Reconhecimento de imagem	Treinamento, implantação

Tabela 7 - Características gerais da plataforma de ML.

As ferramentas suportam uma variedade de tipos de dados, com imagens como o tipo mais frequente (Figura 5). Todas as ferramentas esperam que os usuários

coletem seus próprios dados, fazendo com que os alunos se envolvam de forma criativa com os dados incorporando conjuntos de dados com os quais os alunos podem se relacionar e compreender facilmente, conforme sugerido por Hautea et al. (2017). A coleta de dados é habilitada via webcam, microfone, etc. No entanto, por exemplo, a coleta de até mesmo uma pequena amostra de imagens via webcam requer que os objetos relacionados estejam próximos e pode ser um tanto cansativa depois de um tempo. Outras ferramentas com foco em tipos específicos de entradas permitem que os usuários colem dados de sensores vestíveis. Algumas ferramentas também permitem carregar arquivos diretamente do computador em lote ou individualmente, o que pode ser menos eficiente.

Nome	Tipos de dados	Opções de entrada	Disponibilidade de <i>datasets</i> prontos para o uso
AlpacaML	Movimento	Sensor vestível	--
eCraft2learn	Imagem, postura e som	Upload de arquivos, webcam e microfone	--
<i>Educational Approach to ML with Mobile Applications</i>	Imagem e stream de vídeo	Webcam	--
Milo	Números e texto	Upload de arquivo	Poucos conjuntos de dados populares usados em cursos introdutórios de ML, como o conjunto de dados Iris
PIC (<i>Personal Image Classifier</i>)	Imagem	Upload de arquivo e webcam	--
SnAlp	Agente de jogo	Snap!	--
DeepScratch	Imagem	Webcam	Iris e MNIST

Tabela 8: Características relativas aos dados.

Como back-end de ML, a maioria das ferramentas usa estruturas ou provedores de ML comuns, como Tensorflow ou IBM Watson, ou suas próprias implementações proprietárias (Tabela 9). O predomínio do uso do Tensorflow.js pode ser explicado por sua facilidade de execução, sem a necessidade de instalação do lado do cliente ou de infraestrutura dedicada para a ferramenta. Para acelerar o

treinamento, algumas ferramentas adotam abordagens de aprendizagem por transferência usando MobileNet ou SqueezeNet como modelos de *deep learning* pré-treinados para reconhecimento de imagem, etc.

Em geral, as ferramentas suportam aprendizagem supervisionada, com poucas exceções suportando aprendizagem por reforço (SnAlp) e/ou aprendizagem não supervisionada (Milo). Algumas ferramentas permitem o uso de um servidor em nuvem (eCraft2learn) ou diretamente em um dispositivo móvel (ZHU, 2019). A maioria usa o navegador do usuário para treinar o modelo (PIC, SnAlp, DeepScratch). Como o processo de treinamento do modelo às vezes pode ser lento, as ferramentas que permitem o treinamento localmente podem fazer uso da GPU da máquina local, quando disponível, para acelerar o processo de treinamento (eCraft2Learn).

Nome	Back-end / Algoritmos de ML	Parâmetros do modelo	Tipos de aprendizagem	Parâmetros de treinamento
AlpacaML	Algoritmo DTW	--	Supervisionado	--
eCraft2learn	Modelos de nuvem pré-treinados, suporte a navegador integrado, IBM Watson	Criação de modelo definindo camadas / neurônios, método de otimização, função de perda	Supervisionado	Iterações de treinamento, taxa de aprendizagem, divisão de validação, embaralhamento de dados
<i>Educational Approach to ML with Mobile Applications</i>	Tensorflow.js	--	Supervisionado	--
Milo	Tensorflow.js	Número de recursos, tipo de conexões de camadas, número de nós, função de ativação, função de otimizador	Supervisionado e não supervisionado	Taxa de aprendizagem, função de perda, métricas de treinamento, iterações
PIC (<i>Personal Image Classifier</i>)	Tensorflow.js	Modelo (MobileNet ou Squeezenet), tipo de modelo (convolution, flatten)	Supervisionado	Taxa de aprendizagem, épocas, fração de dados de

		e quantidade de camadas		treinamento, otimizador
SnAlp	Algoritmo Q-Learning	--	Aprendizagem por reforço	Taxa de aprendizagem, fator de desconto, taxa de exploração, ações disponíveis
DeepScratch	Tensorflow.js	Redes neurais: Densa, RNN e CNN	Supervisionado	--

Tabela 9 - Características do modelo de ML e aprendizagem.

Usando ferramentas visuais, os conceitos de ML são normalmente ocultados com caixas pretas para reduzir a carga cognitiva durante o aprendizado (RESNICK et al., 2000). Essas abstrações de conceitos de ML incluem representações de alto nível, pois existem ferramentas (LANE, 2018) onde treinamento do modelo é reduzido a um único botão de ação. No entanto, como essa ocultação dos conceitos de ML limita a capacidade das pessoas de construir uma compreensão básica dos conceitos de ML (HITRON et al., 2019)(ReRESNICKsnick et al., 2000), algumas ferramentas fornecem modos avançados que fornecem uma representação de nível inferior. Por exemplo, eCraft2Learn, Milo e PIC, permitem definir parâmetros da arquitetura da rede neural (como tipo de modelo, número de camadas, etc.). Um modo avançado também é fornecido em relação aos parâmetros de treinamento (como épocas, taxa de aprendizagem, lotes, etc.) como parte do eCraft2Learn, Milo, PIC e SnAIP. No entanto, embora algumas ferramentas forneçam informações breves sobre o vocabulário e/ou esses parâmetros, nenhuma ajuda adicional adaptada ao público-alvo é fornecida para orientar a seleção de seus valores.

Como a incerteza é uma característica inevitável dos modelos de ML na maioria dos aplicativos do mundo real e, portanto, ao interagir com um modelo de ML, é importante que os usuários estejam cientes dessa incerteza para gerenciar suas

expectativas sobre o desempenho do modelo (DUDLEY; KRISTENSSON. 2018). Ainda assim, o conceito de modelo probabilístico e suas limitações podem ser difíceis de transmitir aos alunos, que podem ter dificuldade em compreender as implicações, pois estudos mostram que mesmo um único valor discrepante em um classificador pode resultar em confusão significativa para os usuários (KIM et al., 2015). A este respeito, a maioria das ferramentas também fornece suporte para a avaliação do modelo de ML treinado (Tabela 10), principalmente permitindo que o usuário teste o modelo com novos dados (capturados via webcam, etc.) para os quais o modelo fornece como saída o rótulo previsto ao qual a entrada pertence e o valor de confiança que representa a probabilidade de acerto.

Nome	Métricas de avaliação	Divisão do dataset
AlpacaML	Testando com novas imagens indicando seu rótulo.	Após o treinamento do modelo, os alunos podem adicionar novas ações que os usuários estão realizando para teste.
eCraft2learn	Perda de treinamento, precisão, duração	Divisão manual por usuário
<i>Educational Approach to ML with Mobile Applications</i>	Nível de confiança mostrado como caixas pretas em uma classe.	NI
Milo	Gráfico de precisão, gráfico de perda	NI
PIC (<i>Personal Image Classifier</i>)	Resultados de previsão por imagem, tabela de correção, gráfico de confiança	Após o treinamento do modelo, os alunos podem adicionar novas imagens para teste da mesma forma que na etapa de treinamento.
SnAlp	NI	NI
DeepScratch	função de perda, precisão de treinamento e teste de precisão.	NI

Tabela 10 - Características relativas à avaliação.

Poucas ferramentas visualizam também métricas de desempenho, como precisão e função de perda (eCraft2Learn). Outra abordagem é a visualização de uma tabela de correção e/ou gráfico de confiança (TANG et. al., 2019) (Figura 7). Uma tabela de exatidão mostra todas as imagens de teste e se foram ou não classificadas

corretamente. Isso ajuda os usuários a inferir por que imagens específicas foram classificadas corretamente ou não, comparando as imagens para encontrar semelhanças. Um gráfico de confiança mostra todas as imagens de teste para um rótulo por vez com base na confiança do modelo. Ele permite que os usuários possam inferir as características das imagens que um modelo aprende para rótulos específicos, para que os usuários possam encontrar padrões em como o modelo toma decisões para melhorar sua precisão. Essas visualizações dos resultados da classificação, facilitam a identificação de dados que não estão classificados com precisão e, assim, auxiliam na análise dos alunos para melhorar o desempenho do modelo. O uso de exemplos para apoiar a compreensão das classes parece ser uma solução promissora que ressoa com os usuários (KIM et al., 2015).



Figura 7 - Exemplos de suporte para avaliação dos modelos de ML.

Nenhuma orientação adicional sobre a interpretação das faixas de valores e o nível de desempenho que eles indicam é fornecida, nem dicas sobre como ajustar o modelo se os níveis de desempenho desejados não forem alcançados. Fornecer essas informações como parte de uma ferramenta educacional pode ajudar os alunos a interpretar e compreender os resultados e a orientá-los de forma construtiva para aprender como melhorar o modelo.

3.3.4. Quais são suas características em relação à plataforma de implantação?

Enquanto várias ferramentas fornecem suporte para a implantação como parte de um jogo ou aplicativo móvel, integrado ou como uma extensão de um ambiente de programação baseado em bloco (Figura 8), a solução Milo suporta apenas a exportação do modelo de ML criado.



Figura 8 - Exemplo de suporte de implantação baseado em bloco (PIC).

Dependendo das tarefas específicas que a ferramenta suporta, blocos de programação de ML são fornecidos para incorporar o ML criado ao projeto. De longe, os ambientes baseados em blocos mais adotados são o Scratch, App Inventor (Figura 1) e Snap!. Para incorporar os modelos de ML criados, essas extensões fornecem blocos de programação adicionais. Dependendo da variedade de tarefas suportadas pela ferramenta, isso pode variar de muito poucos blocos (como 3 blocos de reconhecimento de imagem) a conjuntos maiores para diversos fins (Tabela 11). Em geral, esses novos blocos são projetados em conformidade com o design visual da respectiva linguagem de programação baseada em blocos.

Nome	Apenas exportar	Suporte ao <i>Deploy</i>	
		Plataforma	Blocos adicionados para ML
AlpacaML	--	Extensão para Scratch	NI
eCraft2learn	--	Snap!	Blocos diversos para várias tarefas de ML
<i>Educational Approach to ML with Mobile Applications</i>	--	Extensão para App Inventor	Diversas extensões de bloco
Milo	Python code	--	--
PIC (<i>Personal Image Classifier</i>)	--	Extensão para App Inventor e PersonalImageClassifier.aix	A extensão PIC tem três propriedades no designer e 11 blocos no editor de blocos
SnAlp	--	Extensão para Snap!	Diversas extensões de bloco
DeepScratch	--	Extensão para Scratch	8 blocos para uso de ML e 8 blocos para retorno de informações do processo de ML

Tabela 11 - Características da plataforma de implantação.

Dessa forma, as ferramentas permitem que os alunos aprendam os conceitos de ML ao mesmo tempo que os capacita a criar artefatos significativos com impacto direto em suas vidas e comunidades. Isso pode motivá-los a criar aplicativos inovadores que atendam aos seus interesses.

3.3.5. Como as ferramentas foram desenvolvidas e avaliadas?

A maioria das publicações encontradas carece de uma descrição da metodologia de pesquisa adotada para desenvolver as ferramentas de ML. O código-fonte de algumas ferramentas (Tabela 12) está disponível sob licença de código aberto permitindo sua evolução e adaptação.

Nome	Metodologia científica	Disponibilidade do código	Licença do código
AlpacaML	NI	NI	NI
eCraft2learn	NI	https://github.com/ecraft2learn/ai/blobs/master/ecraft2learn.js	BSD
<i>Educational Approach to</i>	NI	NI	NI

<i>ML with Mobile Applications</i>			
Milo	NI	https://miloide.github.io	Apache 2.0 License
PIC (<i>Personal Image Classifier</i>)	NI	https://github.com/mit-cml/appinventor-extensions/tree/extension/personal-image-classifier	Apache-2.0 License
SnAlp	NI	NI	NI
DeepScratch	NI	https://github.com/Noufst/DeepScratch	--

Tabela 12 - Informações sobre o desenvolvimento das ferramentas.

No entanto, outros estudos visando a avaliação das ferramentas são relatados. Os fatores avaliados variam desde a eficácia das ferramentas na aprendizagem, usabilidade, utilidade e eficiência dos alunos até a identificação de seus pontos fortes e fracos. As avaliações foram conduzidas como estudos de caso. Os tamanhos das amostras são em sua maioria pequenos, variando de 6 a 23 participantes, apenas Kahn e Winters apresentaram um estudo com 40 alunos. Os resultados desses estudos indicam que as ferramentas ajudam a alavancar o conhecimento de domínio dos alunos para coletar dados, construir modelos, testar e avaliar modelos, permitindo-lhes realizar iterações rápidas para testar hipóteses sobre o desempenho do modelo e reformular seus modelos. Eles também avaliaram a usabilidade das ferramentas como muito boa. Além disso, as ferramentas parecem ajudar os alunos a desenvolver e discutir teorias sobre como funcionam os modelos e as características de um bom modelo, ajudando-os assim a compreender até mesmo conceitos complexos de ML, bem como a refletir criticamente sobre o impacto do ML na prática. A integração em ambientes gerais de programação baseados em blocos também permite a criação de artefatos atraentes e objetivos.

Name	Fatores de qualidade avaliados	Projeto de pesquisa	Tamanho da amostra	Contexto de aplicação	Descobertas
AlpacaML	Como os jovens podem usar o AlpacaML para construir, testar e refinar modelos de habilidades atléticas que desejam melhorar.	Estudo de caso	6	Alunos de 8 a 14 anos que tiveram experiência com scratch	Aproveita o conhecimento de domínio dos alunos para coletar dados, construir modelos, testar e avaliar modelos; permite que eles conduzam iterações rápidas para testar hipóteses sobre o desempenho do modelo e reformular seus modelos; permite que os alunos desenvolvam teorias sobre como o modelo funciona e as características de um bom modelo.
eCraft2learn	Compreensão do aluno sobre IA e ambiente de agente, percepção e ação, bem como atenção, envolvimento, prazer e usabilidade do ambiente Snap!.	Estudo de caso	0	Alunos do ensino médio e profissionais	77,5% dos alunos indicaram que entendem de IA. Todos os alunos gostaram do processo de aprendizagem, exceto um aluno. Mais de 40% dos alunos afirmaram que foi fácil. 82,5% dos alunos ficaram interessados e motivados em fazer o programa de IA usando o Snap!
<i>Educationa l Approach to ML with Mobile Applications</i>	Aprendizagem e interesse em ML	Estudo de caso pré / pós-teste	0	Alunos do ensino médio	As extensões mostraram-se benéficas para a compreensão dos conceitos introduzidos na primeira parte da aula.
Milo	Utilidade e facilidade de uso da ferramenta, juntamente com o nível percebido de compreensão dos conceitos de ML	Estudo de caso pré / pós-teste	0	Alunos de graduação de um primeiro curso introdutório em ML.	90% dos participantes relataram que as visualizações eram muito fáceis de criar usando o Milo e complementam sua compreensão dos conceitos. 70% dos alunos consideraram que a ferramenta seria muito útil para alunos iniciantes.
PIC (<i>Personal Image Classifier</i>)	Usabilidade das ferramentas PIC e sua eficácia na introdução de ML aos novatos	Estudo de caso pré / pós-teste	3	Oficinas com alunos do ensino médio.	Os alunos gostaram de usar a interface PIC e o aplicativo Expressions Match, indicando que as ferramentas eram intuitivas e divertidas de usar. Os alunos puderam usar as ferramentas de análise para desenvolver o raciocínio sobre como seus modelos estavam se comportando. A maneira

					como a ferramenta fornece representações visuais de discussões guiadas habilitadas por dados sobre o desequilíbrio do conjunto de dados e como isso pode levar ao que parece ser um modelo tendencioso.
SnAlp	--	--	-	--	--
DeepScratch	Teste de funcionalidade e usabilidade da ferramenta DeepScratch	Estudo de caso	5	Alunos 8 a 17 anos	Apesar da pequena amostragem, os autores relatam que atingiram um resultado satisfatório ao conseguir ensinar ML ao grupo aplicado.

Tabela 13: Informações sobre a avaliação das ferramentas.

3.4. Discussão

Atualizando a pesquisa feita por Gresse von Wangenheim *et al.* (2020) e filtrando os resultados apenas que abordavam o ensino médio foi possível identificar um total de 7 ferramentas sendo utilizadas para ensinar ML neste contexto. Por estarem em ambientes visuais as ferramentas acabam se tornando mais atrativas, pois a interação com os usuários acaba acontecendo de forma mais intuitiva uma vez que estes ambientes possuem um design mais simples e atraente que ambientes de programação tradicionais. As ferramentas permitem a exploração por parte dos alunos, possibilitando a criação de diferentes modelos personalizados de ML. Ao permitir a customização em geral do processo de ML através de uma interface visual é possível que os alunos desenvolvam interativamente seus modelos, permitindo que a cada iteração os mesmos avaliem o estado atual do desenvolvimento e que tomem as ações necessárias para continuar a melhorá-lo, desenvolvendo assim suas capacidades e seu pensamento a respeito de ML.

Na pesquisa foi possível observar que é possível que os jovens aprendam conceitos de ML na prática em um ambiente sem código. Os ambientes baseados em workflow se mostraram eficientes ao levar os alunos do ensino médio por um caminho mais guiado, mostrando passo a passo todas as etapas necessárias para o desenvolvimento do modelo, estando bem alinhado com seu processo cognitivo. Os ambientes em blocos por sua vez se mostraram um pouco mais complexos para o desenvolvimento do processo de ML em geral. Em contrapartida apresentaram uma maior capacidade para o desenvolvimento de modelos mais sofisticados poderosos o suficiente para soluções mais completas ao custo de que os alunos já tenham uma compreensão prévia dos conceitos e práticas de computação.

Encontrar o equilíbrio entre quais conceitos de ML devem ser abordados com os alunos do ensino médio requer atenção, pois os mesmos não têm educação formal obrigatória a respeito do assunto. É muito importante que os ambientes não sejam uma “caixa preta” onde os alunos insiram os dados e recebam um modelo pronto, uma vez que dessa forma acaba não ensinando os conceitos de ML em si. Em geral, todas as ferramentas atendem bem a isso, passando por boa parte dos conceitos de ML e dando as opções de customizações necessárias quanto a atividade sendo executada. Ao permitir que os alunos tenham a possibilidade de criar seus próprios, se abre uma grande oportunidade para que eles explorem cada vez mais as áreas de IA/ML, além de contribuir com seu desenvolvimento de sua criatividade. Entretanto as ferramentas não oferecem um suporte educacional mais abrangente como o fornecimento de instruções e dicas construtivas, prática comum em ambientes que se propõem a ensinar programação a jovens.

Todas as ferramentas estão disponíveis online e com acesso pelo navegador, facilitando o acesso às mesmas uma vez que não se fica preso ao ambiente com

dependências de bibliotecas por exemplo, entretanto a conexão contínua com a internet pode ser um problema em alguns cenários. Em geral as ferramentas são acompanhadas por um conjunto de recursos educacionais que vão de tutoriais até alguns cursos extracurriculares de curta duração, porém estão disponíveis apenas em inglês, o que pode inviabilizar a aprendizagem dos alunos do ensino médio.

Ameaças à validade: Ameaças à validade. A fim de minimizar as ameaças à validade dos resultados deste estudo, foram identificadas ameaças potenciais e foram aplicadas estratégias de mitigação. As revisões sistemáticas sofrem com o viés comum de que resultados positivos têm maior probabilidade de serem publicados do que resultados negativos. No entanto, não consideramos isso uma ameaça crítica à esta pesquisa, pois em vez de focar no impacto dessas ferramentas, foi buscado caracterizar as próprias ferramentas. Para mitigar a omissão de estudos relevantes, foi construída cuidadosamente a *string* de pesquisa para ser o mais inclusivo possível, considerando não apenas os conceitos principais, mas também os sinônimos. O risco de excluir estudos primários relevantes foi ainda mais atenuado pelo uso de vários bancos de dados e a inclusão de literatura secundária. Ameaças à seleção de estudos e extração de dados foram mitigadas através do fornecimento de uma definição detalhada dos critérios de inclusão/exclusão.

4. *Visual Layer Classification* para o desenvolvimento de modelo de classificação de imagens

Este capítulo apresenta o desenvolvimento do *Visual Layer Classification* (Vulcan) além da aplicação prática para a classificação do modelo de árvores nativas.

4.1. Análise de Requisitos

O objetivo deste trabalho é desenvolver uma interface *visual compatível* com *Jupyter Notebooks* no *Google Colab* para a tarefa de classificação de imagens *single label*. A interface será baseada no processo centrado no ser humano (GRESSE VON WANGENHEIM; VON WANGENHEIM, 2021) desde a análise de requisitos, entrada dos dados já classificados até a exportação do modelo treinado.

A ferramenta deve se conectar ao Google Drive para obter o conjunto de dados previamente separado em pastas com as classes. Além disso, na etapa de predição deverá ser possível realizar o *upload* de uma imagem para realizar a predição. Serão aceitas imagens de diferentes tamanhos e formatos uma vez que estes detalhes serão tratados internamente.

A respeito do processo de *ML* a ferramenta deve contemplar o tratamento de imagens como conversão e redimensionamento além de também suportar o processo de aumento, caso desejado. As etapas de treinamento dos modelos de *ML* permitirão a seleção da arquitetura do modelo, além alguns hiperparâmetros como o número de épocas, podendo também realizar o fine tuning do modelo. A visualização de medidas de desempenho como por exemplo acurácia, *error rate*, matriz de

confusão também deverá ser apresentada. Ao final do processo, a ferramenta deverá permitir a exportação do modelo.

Ao final do desenvolvimento, a interface será aplicada ao contexto da tarefa real de classificação de espécie de árvores.

4.2. Arquitetura do sistema

A Figura 9 mostra a visão geral do Vulcan, que consiste em um ambiente que rode sobre Jupyter Notebooks, uma vez que um dos objetivos é popularizar e democratizar o acesso a IA/ML e os Jupyter Notebooks são acessíveis diretamente pelo navegador do computador, independentes do sistema operacional e sem a necessidade de instalação de programas e/ou bibliotecas adicionais. Como principal opção para viabilizar o uso deste ambiente tem-se o Google Colab, onde é possível utilizar a plataforma de forma gratuita além de permitir utilizar GPUs.

A entrada do dataset é feita por meio do Google Drive, uma vez que a integração com o Colab é suportada e de fácil desenvolvimento e, além disso, é possível gerenciar, visualizar e compartilhar o dataset facilmente.

Para os elementos visuais é utilizada a biblioteca *Jupyter Widgets* (IPYWIDGETS, 2021) que é a principal referência em elementos visuais utilizados em *Jupyter Notebooks*.

Para a tarefa de classificação de imagens é utilizado a biblioteca *fast.ai*. Ela é uma biblioteca de IA/ML com a curva de aprendizado mais rápida em comparação a outras bibliotecas (FAST.AI, 2021).

Apesar de suportar elementos visuais, o Colab não foi concebido como uma interface gráfica ou aplicação de uma célula. Normalmente se utilizam diversas células, uma com cada trecho de código referente a etapa do processo de ML. Entretanto é possível encapsular todos os processos de IA/ML de acordo com os elementos visuais disponíveis e executar a aplicação de ponta a ponta em apenas uma célula.

É utilizado um pacote PIP para a distribuição do Visual Layer Classification para ser disponibilizado no Notebook com o mínimo de linhas de código.

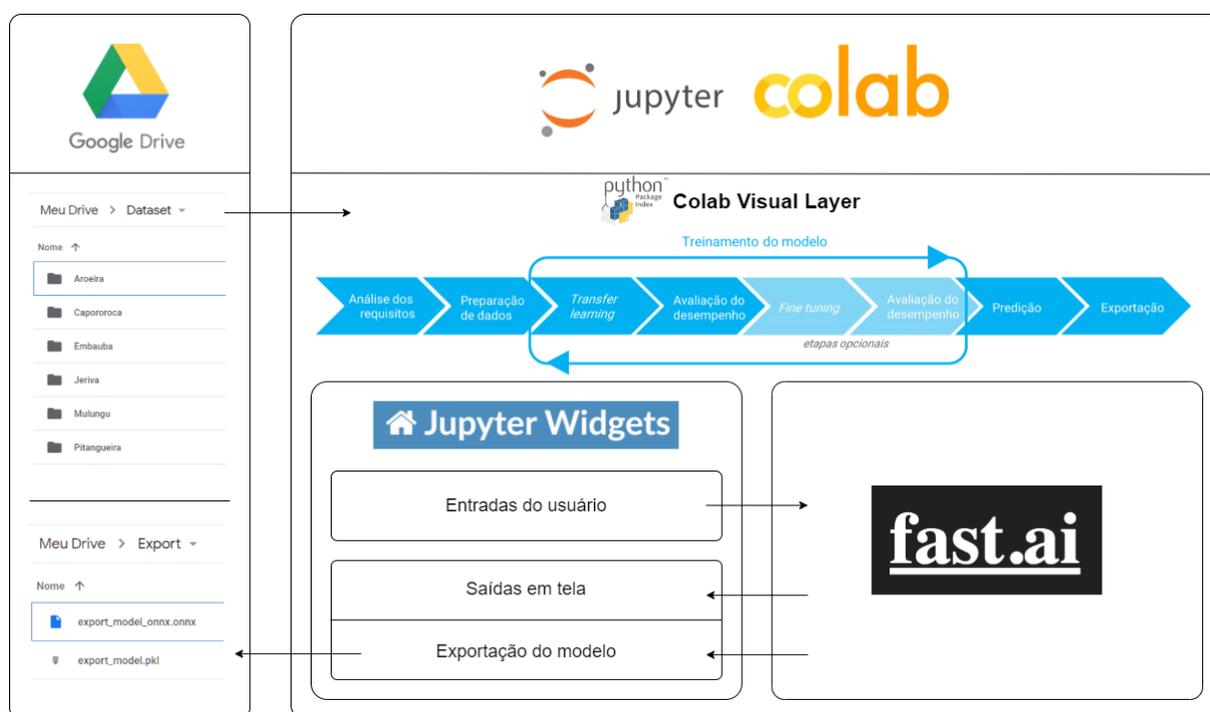


Figura 9 - Visão geral do Vulcan

4.3. Implementação do Vulcan

Nesta seção é apresentado o desenvolvimento e implantação da biblioteca. Toda a estrutura proposta foi desenvolvida utilizando Python (PYTHON, 2021) por rodar nativamente no colab e ter total compatibilidade com os frameworks selecionados.

De acordo com a análise de requisitos e a arquitetura definida seguindo o human-centric interactive ML process (GRESSE VON WANGENHEIM; VON WANGENHEIM, 2021) foram especificadas as necessidades de entrada e saída de dados guiando o suporte desenvolvido *Visual Layer Classification* conforme apresentado na Tabela 14.

Fase do processo de ML	Passo do processo	Detalhes e padrões
Análise de requisitos	Especificação da(s) tarefa(s) e requisitos do sistema de ML	<ul style="list-style-type: none"> • Tarefa • Tipo da tarefa • Categorias • Fonte de dados • Quantidade de dados • Padronização das imagens • Rotulação dos dados • Desempenho
Preparação de dados	Conjunto de dados (no <i>Google Drive</i>)	Um conjunto de dados de imagens deve ser disponibilizado em uma pasta no Google Drive e dentro desta pasta deve haver uma pasta para cada label desejado. As imagens ficarão dentro da pasta de cada label respectivo.
	Definição do conjunto de dados	Caminho do diretório no Google Drive
	Visualizar informações do conjunto de dados	Percorre o conjunto de dados informados e mostra um histograma com a distribuição das imagens nas classes utilizando a função
	Padronização das imagens: Redimensionamento	Padrão = Sim <ul style="list-style-type: none"> • Sim <ul style="list-style-type: none"> - Tamanho: 224px - Método redimensionamento: "squish" • Não

		<ul style="list-style-type: none"> - Tamanho: (min=8px, max=640px) - Método redimensionamento: ("crop", "pad", "squish")
Padronização das imagens: Transformações e aumentações	Padrão = Desabilitado	<ul style="list-style-type: none"> • Desabilitado - Não realiza nenhuma alteração • Habilitado - Cria aumentações das imagens de acordo com as transformações: <ul style="list-style-type: none"> - Espelhar horizontalmente: (Sim, Não) Padrão = Sim - Espelhar verticalmente: (Sim, Não) Padrão = Não - Zoom máximo: (min=1.0,max=4) Padrão = 1.1 - Zoom mínimo: (min=0.5,max=1) Padrão = 0.9 - Iluminação máxima: (min=0.2,max=1) Padrão = 0.2 - Intensidade das transformações: (min = 0,max = 3) Padrão = 1 - Rotação máxima: (min=0,max=30) Padrão = 10 - Intensidade iluminação: (min=0.25, max=1) Padrão = 0.75 - Inclinação: (min=0.1, max=1) Padrão = 0.2
Separação do conjunto de dados de treinamento e validação	Padrão = 20%	
Tamanho do Batch	Padrão = 16	<ul style="list-style-type: none"> • 8 • 16 • 32 • 64 • 128 • 256
Criar dataloader	Cria o objeto que agrupa os elementos definidos anteriormente (dataset, redimensionamento, transformações e aumentações (quando houver), splitter e o tamanho do batch). Além disso, converte todas as imagens do dataset para o mesmo formato (jpg) e padroniza os nomes de acordo com a pasta.	
Visualizar batch	<ul style="list-style-type: none"> • Qtd. (itens do batch): (min = 1,max = 15) Padrão = 3 • Somente variações da mesma imagem?: <ul style="list-style-type: none"> - Sim: Exibe as aumentações e transformações de apenas uma imagem - Não: Mostra as diferentes imagens do batch Padrão = Não	

Transfer Learning	Seleção da arquitectura	<p>Padrão = resnet18</p> <ul style="list-style-type: none"> • alexnet • densenet121 • densenet161 • densenet169 • densenet201 • resnet18 • resnet34 • resnet50 • resnet101 • resnet152 • squeezeenet1_0 • squeezeenet1_1 • vgg16_bn • vgg19_bn
	Métricas do acompanhamento do treinamento	<p>Padrão = Acurácia</p> <ul style="list-style-type: none"> • Taxa de erro • Acurácia
	Criar Learner	Cria o objeto que irá "aprender". Será criado partir dos elementos definidos anteriormente no dataloader em conjunto com a arquitetura e as métricas definidas
	Epochs	<p>Padrão = 10</p> <ul style="list-style-type: none"> • min=1 • max=50
	Treinar modelo	Executa o treinamento do modelo a partir do objeto learner com a quantidade de epochs informada.
	Data Cleaning	Permite visualizar a distribuição dos arquivos contidos no learner e efetuar correções e/ou customizações caso necessário.
	Avaliação do Transfer Learning	Matriz de confusão
Acurácia por categoria		Exibe a acurácia por categoria além de outras métricas de desempenho do treinamento do modelo como f1 score
Principais perdas		<p>Padrão = 3</p> <p>Exibe até 10 principais perdas do modelo treinado exibindo</p> <ul style="list-style-type: none"> • Imagem • Classificação original • Classificação predita • Métricas
Mais Confundidos		<p>Padrão = 3</p> <p>Exibe a informação textual de quais itens foram confundidos e quantas vezes foram confundidos de acordo com o valor informado.</p>
Fine tuning (opcional)		Descongelar modelo
	Encontrar melhor taxa	Executa a busca das melhores taxas de aprendizagem para o modelo e salva esse valor, além de exibir em tela as taxas: min, step, valley e slide.
	Epochs fine tuning	<p>Padrão = 5</p> <ul style="list-style-type: none"> • min=1 • max=50

	Treinar modelo otimizado	Padrão = 10 <ul style="list-style-type: none"> • min=1 • max=50
Avaliação do Fine tuning	Matriz de confusão	Exibe a matriz de confusão do modelo treinado
	Acurácia por categoria	Exibe a acurácia por categoria além de outras métricas de desempenho do treinamento do modelo como f1 score
	Principais perdas	Padrão = 3 Exibe até 10 principais perdas do modelo treinado exibindo <ul style="list-style-type: none"> • Imagem • Classificação original • Classificação predita • Métricas
	Mais Confundidos	Padrão = 3 Exibe a informação textual de quais itens foram confundidos e quantas vezes foram confundidos de acordo com o valor informado.
Predição	<i>Upload</i>	Permite o upload de uma imagem (por vez) para realizar a predição da mesma.
	Predição	Exibe a imagem e o resultado da predição com a categoria resultante com percentual de confiança.
Exportação	Exportar modelo PKL	Exporta o modelo no diretório informado (da mesma forma que o path) usando pickle_protocol.
	Exportar modelo ONNX	Exporta o modelo no diretório informado (da mesma forma que o path) no formato ONNX (Open Neural Network Exchange), permitindo assim interoperabilidade entre os principais frameworks.

Tabela 14 - Atividades suportadas pelo Vulcan

4.3.1. Integração com o Google Drive

A integração com o Google Drive (COLAB, 2021) é a ponte entre a entrada e saída dos dados com o Visual Layer Classification, permitindo tanto a entrada do conjunto de dados para o treinamento do modelo quanto a exportação do modelo treinado.

A autenticação é feita com a API da Google e é possível logar com a conta que desejar (não necessariamente a mesma do Colab), esta conta se conectará ao drive apenas nas pastas informadas pelo próprio usuário durante a execução.

4.3.2. Biblioteca *Fast.ai*

São utilizados uma série de objetos da biblioteca do fast.ai para realizar o processo completo de ML. São utilizados os seguintes objetos da biblioteca:

- Path, objeto que guarda o caminho dos dados originais;
- Dataloader, objeto que agrupa além das imagens, informações sobre as transformações, aumentações, callbacks, splitter e o tamanho do batch;
- Learner, objeto criado a partir do dataloader, nele sendo definida a arquitetura em que o modelo será treinado bem como as métricas de treinamento disponíveis (taxa de erro e acurácia);
- Interp, objeto criado após o learner terminar o treinamento (seja o inicial ou o fine-tuning), contendo as informações relativas ao modelo e o treinamento do modelo

4.3.3. Processo de *ML*

O fluxo do processo de ML no Visual Layer Classification foi desenvolvido seguindo o processo de desenvolvimento de ML centrado no ser humano que pode ser visto na Figura 10 (GRESSE VON WANGENHEIM; VON WANGENHEIM, 2021), e também nas necessidades da biblioteca do fast.ai apresentadas na seção anterior. Para fazer essa camada intermediária foram criadas algumas funções para garantir os objetos necessários além da execução das etapas pré-definidas.



Figura 10 - Etapas do processo de ML centrado no ser humano

`build_path(path_f)`: Responsável pela variável global que guarda o path contendo o conjunto de dados para ser utilizado no restante da ferramenta.

`build_data(path_fun, splitter_percent_validation, splitter_bs)`: Responsável pela variável global que guarda o *dataloader* para ser utilizado no restante da ferramenta. Para a criação do *dataloader* anteriormente são criados os seguintes itens de acordo com os parâmetros recebidos: *splitter*, redimensionamento, transformações e aumentações (caso desejado) e *datablock*.

`create_learner(model_architecture)`: Responsável pela variável global que guarda o *learner* para ser utilizado no restante da ferramenta. Para a criação do *learner* anteriormente é criado o *callback* de acordo com as métricas selecionadas na interface.

`train_data(epochs)`: Responsável pela variável global que guarda o *interp* para ser utilizado no restante da ferramenta. Treina o modelo de acordo com número de *epochs* selecionada na interface

`jpg_converter(path)`: Responsável por converter os arquivos para *jpg* e renomear os mesmos com o prefixo da pasta seguido de um número sequencial

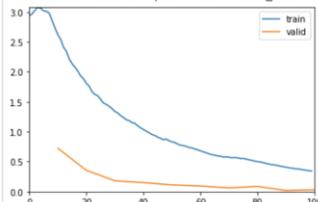
4.3.4. Interface visual

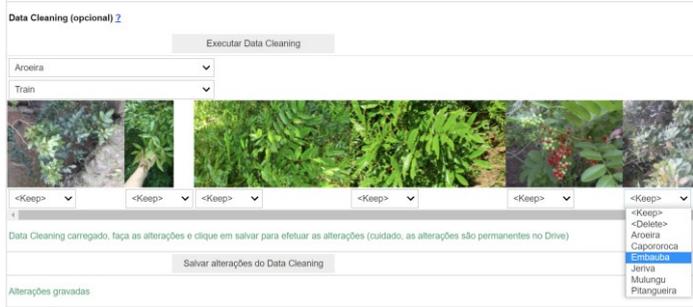
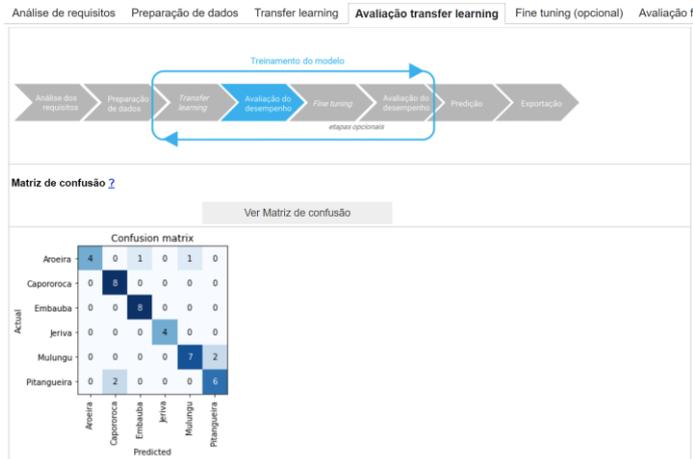
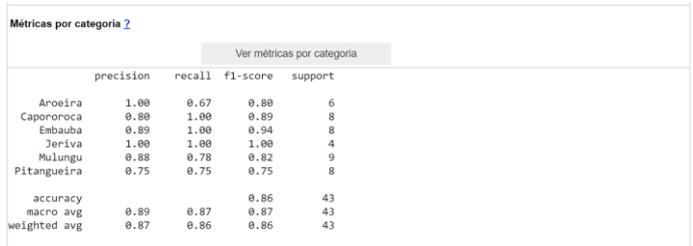
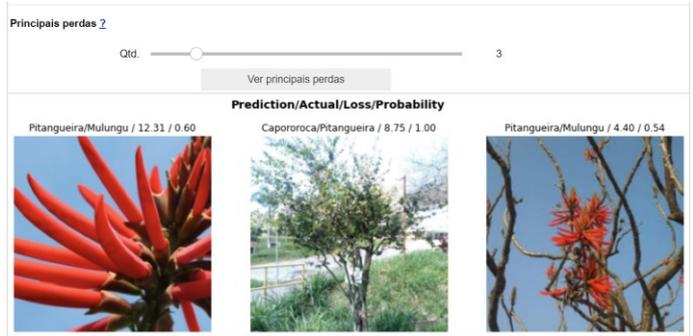
A interface representa visualmente todo o processo de ML/IA na ferramenta, integrando todos os elementos. Para criar um ambiente sem código, tudo foi encapsulado na função `visual_layer()`.

A base interface visual da ferramenta foi criada com a biblioteca do Google Colab widgets. Foram utilizados tabs para criação das abas e grids para conter os visuais e saídas da ferramenta. Para os demais elementos (textos, entradas e botões) foi utilizado a biblioteca ipywidgets com seus diversos widgets, styles, layouts e interações. A Tabela 15 mostra o resultado da interface visual desenvolvida.

Fase do processo de ML	Passo de processo	Interface
Análise de requisitos	Especificação da(s) tarefa(s) e requisitos do sistema de ML	

Preparação de dados	Definição do conjunto de dados	<p>Análise de requisitos Preparação de dados Transfer learning Avaliação transfer learning Fine tuning (opcional)</p> <p>Definição do Path ?</p> <p>/Treinamento Carregar Path</p> <p>Path: /content/gdrive/My Drive/Treinamento/ definido com sucesso</p>												
	Visualizar informações do conjunto de dados	<p>Visualizar informações do conjunto de dados ?</p> <p style="text-align: center;">Ver total por categoria</p> <table border="1"> <caption>Quantidade por categoria</caption> <thead> <tr> <th>Categoria</th> <th>Quantidade</th> </tr> </thead> <tbody> <tr> <td>Aroeira</td> <td>31</td> </tr> <tr> <td>Mulungu</td> <td>45</td> </tr> <tr> <td>Embauba Pitangueira</td> <td>31</td> </tr> <tr> <td>Jeriva</td> <td>29</td> </tr> <tr> <td>Capororoca</td> <td>45</td> </tr> </tbody> </table>	Categoria	Quantidade	Aroeira	31	Mulungu	45	Embauba Pitangueira	31	Jeriva	29	Capororoca	45
	Categoria	Quantidade												
	Aroeira	31												
	Mulungu	45												
	Embauba Pitangueira	31												
	Jeriva	29												
Capororoca	45													
Padronização das imagens: Redimensionamento	<p>Redimensionamento ?</p> <p><input type="checkbox"/> Padrão</p> <p>Tamanho das imagens <input type="range" value="224"/> 224</p> <p>Método redimensionamento crop</p>													
Padronização das imagens: Transformações e aumentações	<p>Transformações e Aumentações ?</p> <p><input type="checkbox"/> Desabilitar</p> <p>Zoom máximo <input type="range" value="1.10"/> 1.10</p> <p>Zoom mínimo <input type="range" value="0.90"/> 0.90</p> <p>Espelhar horizontalmente Sim Não</p> <p>Espelhar verticalmente Sim Não</p> <p>% de intensidade <input type="range" value="1.00"/> 1.00</p> <p>Rotação máxima <input type="range" value="10"/> 10</p> <p>Iluminação máxima <input type="range" value="0.20"/> 0.20</p> <p>Intensidade iluminação <input type="range" value="0.75"/> 0.75</p> <p>Inclinação <input type="range" value="0.20"/> 0.20</p>													
Separação do conjunto de dados de treinamento e validação	<p>Separação do conjunto de dados de treinamento e validação ?</p> <p>% Validação <input type="range" value="0.20"/> 0.20</p>													
Tamanho do Batch	<p>Criação do dataloader ?</p> <p>Tamanho do batch <input type="range" value="16"/> 16</p> <p style="text-align: center;">Criar dataloader</p> <p>Dataloader criado com sucesso</p>													
Criar dataloader														

	<p>Visualizar batch</p>																																																																																								
<p>Transfer Learning</p>	<p>Seleção da arquitetura</p>	<p>Análise de requisitos Preparação de dados Transfer learning Avaliação transfer learning Fine tuning (opc)</p>																																																																																							
	<p>Métricas do acompanhamento do treinamento</p>																																																																																								
	<p>Criar Learner</p>	<p>Arquitetura ?</p> <ul style="list-style-type: none"> resnet18 alexnet densenet121 densenet161 densenet169 densenet201 resnet18 resnet34 resnet50 resnet101 resnet152 squeezenet1_0 squeezenet1_1 vgg16_bn vgg19_bn <p>Métricas ?</p> <p><input type="checkbox"/> Taxa de erro <input type="checkbox"/> Acurácia</p> <p>Criar learner</p> <p>Learner criado com sucesso</p>																																																																																							
<p>Epochs</p>																																																																																									
<p>Treinar modelo</p>	<p>Treinamento do modelo ?</p> <p>Epochs <input type="range" value="10"/></p> <p>Treinar modelo</p> <table border="1"> <thead> <tr> <th>epoch</th> <th>train_loss</th> <th>train_error_rate</th> <th>train_accuracy</th> <th>valid_loss</th> <th>valid_error_rate</th> <th>valid_accuracy</th> <th>time</th> </tr> </thead> <tbody> <tr><td>0</td><td>2.735731</td><td>0.725000</td><td>0.275000</td><td>1.555996</td><td>0.488372</td><td>0.511628</td><td>01:00</td></tr> <tr><td>1</td><td>1.885558</td><td>0.356250</td><td>0.643750</td><td>1.310523</td><td>0.348837</td><td>0.651163</td><td>00:38</td></tr> <tr><td>2</td><td>1.386816</td><td>0.181250</td><td>0.818750</td><td>1.164684</td><td>0.279070</td><td>0.720930</td><td>00:38</td></tr> <tr><td>3</td><td>1.066164</td><td>0.150000</td><td>0.850000</td><td>1.017325</td><td>0.162791</td><td>0.837209</td><td>00:38</td></tr> <tr><td>4</td><td>0.851594</td><td>0.112500</td><td>0.887500</td><td>1.058274</td><td>0.209302</td><td>0.790698</td><td>00:38</td></tr> <tr><td>5</td><td>0.699028</td><td>0.093750</td><td>0.906250</td><td>0.797181</td><td>0.139535</td><td>0.860465</td><td>00:38</td></tr> <tr><td>6</td><td>0.582211</td><td>0.062500</td><td>0.937500</td><td>0.845111</td><td>0.186046</td><td>0.813953</td><td>00:37</td></tr> <tr><td>7</td><td>0.511068</td><td>0.087500</td><td>0.912500</td><td>0.802185</td><td>0.209302</td><td>0.790698</td><td>00:38</td></tr> <tr><td>8</td><td>0.417291</td><td>0.018750</td><td>0.981250</td><td>0.814731</td><td>0.209302</td><td>0.790698</td><td>00:38</td></tr> <tr><td>9</td><td>0.343687</td><td>0.031250</td><td>0.968750</td><td>0.852464</td><td>0.209302</td><td>0.790698</td><td>00:38</td></tr> </tbody> </table> <p>Better model found at epoch 0 with valid_loss value: 1.555996060371399. Better model found at epoch 0 with valid_loss value: 1.555996060371399.</p>  <p>Better model found at epoch 1 with valid_loss value: 1.3105233907699585. Better model found at epoch 1 with valid_loss value: 1.3105233907699585. Better model found at epoch 2 with valid_loss value: 1.1646844148635864. Better model found at epoch 2 with valid_loss value: 1.1646844148635864. Better model found at epoch 3 with valid_loss value: 1.0173251628875732. Better model found at epoch 3 with valid_loss value: 1.0173251628875732. Better model found at epoch 5 with valid_loss value: 0.7971811294555664. Better model found at epoch 5 with valid_loss value: 0.7971811294555664.</p> <p>Fim do treinamento do modelo</p>	epoch	train_loss	train_error_rate	train_accuracy	valid_loss	valid_error_rate	valid_accuracy	time	0	2.735731	0.725000	0.275000	1.555996	0.488372	0.511628	01:00	1	1.885558	0.356250	0.643750	1.310523	0.348837	0.651163	00:38	2	1.386816	0.181250	0.818750	1.164684	0.279070	0.720930	00:38	3	1.066164	0.150000	0.850000	1.017325	0.162791	0.837209	00:38	4	0.851594	0.112500	0.887500	1.058274	0.209302	0.790698	00:38	5	0.699028	0.093750	0.906250	0.797181	0.139535	0.860465	00:38	6	0.582211	0.062500	0.937500	0.845111	0.186046	0.813953	00:37	7	0.511068	0.087500	0.912500	0.802185	0.209302	0.790698	00:38	8	0.417291	0.018750	0.981250	0.814731	0.209302	0.790698	00:38	9	0.343687	0.031250	0.968750	0.852464	0.209302	0.790698	00:38
epoch	train_loss	train_error_rate	train_accuracy	valid_loss	valid_error_rate	valid_accuracy	time																																																																																		
0	2.735731	0.725000	0.275000	1.555996	0.488372	0.511628	01:00																																																																																		
1	1.885558	0.356250	0.643750	1.310523	0.348837	0.651163	00:38																																																																																		
2	1.386816	0.181250	0.818750	1.164684	0.279070	0.720930	00:38																																																																																		
3	1.066164	0.150000	0.850000	1.017325	0.162791	0.837209	00:38																																																																																		
4	0.851594	0.112500	0.887500	1.058274	0.209302	0.790698	00:38																																																																																		
5	0.699028	0.093750	0.906250	0.797181	0.139535	0.860465	00:38																																																																																		
6	0.582211	0.062500	0.937500	0.845111	0.186046	0.813953	00:37																																																																																		
7	0.511068	0.087500	0.912500	0.802185	0.209302	0.790698	00:38																																																																																		
8	0.417291	0.018750	0.981250	0.814731	0.209302	0.790698	00:38																																																																																		
9	0.343687	0.031250	0.968750	0.852464	0.209302	0.790698	00:38																																																																																		

	<p>Data Cleaning</p>																																																			
	<p>Matriz de confusão</p>	 <p>Confusion matrix</p> <table border="1"> <thead> <tr> <th>Actual \ Predicted</th> <th>Aroeira</th> <th>Capororoca</th> <th>Embauba</th> <th>Jeriva</th> <th>Mulungu</th> <th>Pitangueira</th> </tr> </thead> <tbody> <tr> <th>Aroeira</th> <td>4</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <th>Capororoca</th> <td>0</td> <td>6</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>Embauba</th> <td>0</td> <td>0</td> <td>8</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>Jeriva</th> <td>0</td> <td>0</td> <td>0</td> <td>4</td> <td>0</td> <td>0</td> </tr> <tr> <th>Mulungu</th> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>7</td> <td>2</td> </tr> <tr> <th>Pitangueira</th> <td>0</td> <td>2</td> <td>0</td> <td>0</td> <td>0</td> <td>6</td> </tr> </tbody> </table>	Actual \ Predicted	Aroeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira	Aroeira	4	0	1	0	1	0	Capororoca	0	6	0	0	0	0	Embauba	0	0	8	0	0	0	Jeriva	0	0	0	4	0	0	Mulungu	0	0	0	0	7	2	Pitangueira	0	2	0	0	0	6	
Actual \ Predicted	Aroeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira																																														
Aroeira	4	0	1	0	1	0																																														
Capororoca	0	6	0	0	0	0																																														
Embauba	0	0	8	0	0	0																																														
Jeriva	0	0	0	4	0	0																																														
Mulungu	0	0	0	0	7	2																																														
Pitangueira	0	2	0	0	0	6																																														
<p>Avaliação do Transfer Learning</p>	<p>Acurácia por categoria</p>	 <p>Métricas por categoria</p> <table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Aroeira</td> <td>1.00</td> <td>0.67</td> <td>0.80</td> <td>6</td> </tr> <tr> <td>Capororoca</td> <td>0.80</td> <td>1.00</td> <td>0.89</td> <td>8</td> </tr> <tr> <td>Embauba</td> <td>0.89</td> <td>1.00</td> <td>0.94</td> <td>8</td> </tr> <tr> <td>Jeriva</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>4</td> </tr> <tr> <td>Mulungu</td> <td>0.88</td> <td>0.78</td> <td>0.82</td> <td>9</td> </tr> <tr> <td>Pitangueira</td> <td>0.75</td> <td>0.75</td> <td>0.75</td> <td>8</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.86</td> <td>43</td> </tr> <tr> <td>macro avg</td> <td>0.89</td> <td>0.87</td> <td>0.87</td> <td>43</td> </tr> <tr> <td>weighted avg</td> <td>0.87</td> <td>0.86</td> <td>0.86</td> <td>43</td> </tr> </tbody> </table>		precision	recall	f1-score	support	Aroeira	1.00	0.67	0.80	6	Capororoca	0.80	1.00	0.89	8	Embauba	0.89	1.00	0.94	8	Jeriva	1.00	1.00	1.00	4	Mulungu	0.88	0.78	0.82	9	Pitangueira	0.75	0.75	0.75	8	accuracy			0.86	43	macro avg	0.89	0.87	0.87	43	weighted avg	0.87	0.86	0.86	43
	precision	recall	f1-score	support																																																
Aroeira	1.00	0.67	0.80	6																																																
Capororoca	0.80	1.00	0.89	8																																																
Embauba	0.89	1.00	0.94	8																																																
Jeriva	1.00	1.00	1.00	4																																																
Mulungu	0.88	0.78	0.82	9																																																
Pitangueira	0.75	0.75	0.75	8																																																
accuracy			0.86	43																																																
macro avg	0.89	0.87	0.87	43																																																
weighted avg	0.87	0.86	0.86	43																																																
	<p>Principais perdas</p>	 <p>Principais perdas</p> <p>Qtd. <input type="range" value="3"/> 3</p> <p>Prediction/Actual/Loss/Probability</p> <ul style="list-style-type: none"> Pitangueira/Mulungu / 12.31 / 0.60 Capororoca/Pitangueira / 8.75 / 1.00 Pitangueira/Mulungu / 4.40 / 0.54 																																																		
	<p>Mais Confundidos</p>	 <p>Mais confundidos</p> <p>vez(es) confundido <input type="range" value="2"/> 2</p> <p>Mulungu foi confundida com Pitangueira e esta combinação em particular ocorreu 2 vez(es) Pitangueira foi confundida com Capororoca e esta combinação em particular ocorreu 2 vez(es)</p>																																																		

Fine tuning (opcional)	Descongelo do modelo	<p>Análise de requisitos Preparação de dados Transfer learning Avaliação transfer learning Fine tuning (opcional) Av.</p> <p>Encontrar melhor taxa de aprendizagem ?</p> <p>Descongelo do modelo</p> <p>Encontrar melhor taxa</p> <p>Modelo descongelado com sucesso</p> <p>lr_min: 1.58489319881447e-07 lr_steep: 3.981071586167673e-06 lr_valley: 5.248074739938602e-05 lr_slide: 7.585775892948732e-05</p> <p>Find lr executado com sucesso</p>																																																																										
	Encontrar melhor taxa	<p>Epochs fine tuning ?</p> <p>Epochs <input type="range" value="5"/></p> <p>Treinar modelo otimizado</p> <table border="1"> <thead> <tr> <th>epoch</th> <th>train_loss</th> <th>train_error_rate</th> <th>train_accuracy</th> <th>valid_loss</th> <th>valid_error_rate</th> <th>valid_accuracy</th> <th>time</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0.136299</td> <td>0.037500</td> <td>0.962500</td> <td>0.512523</td> <td>0.162791</td> <td>0.837209</td> <td>00:39</td> </tr> <tr> <td>1</td> <td>0.110495</td> <td>0.031250</td> <td>0.968750</td> <td>0.626146</td> <td>0.209302</td> <td>0.790698</td> <td>00:39</td> </tr> <tr> <td>2</td> <td>0.157506</td> <td>0.068750</td> <td>0.931250</td> <td>0.798249</td> <td>0.209302</td> <td>0.790698</td> <td>00:40</td> </tr> <tr> <td>3</td> <td>0.155925</td> <td>0.056250</td> <td>0.943750</td> <td>0.527339</td> <td>0.162791</td> <td>0.837209</td> <td>00:40</td> </tr> <tr> <td>4</td> <td>0.137334</td> <td>0.043750</td> <td>0.956250</td> <td>0.561539</td> <td>0.139535</td> <td>0.860465</td> <td>00:39</td> </tr> </tbody> </table> <p>Better model found at epoch 0 with valid_loss value: 0.5125232934951782. Better model found at epoch 0 with valid_loss value: 0.5125232934951782.</p> <p>Treino fine tuning executado com sucesso</p>	epoch	train_loss	train_error_rate	train_accuracy	valid_loss	valid_error_rate	valid_accuracy	time	0	0.136299	0.037500	0.962500	0.512523	0.162791	0.837209	00:39	1	0.110495	0.031250	0.968750	0.626146	0.209302	0.790698	00:39	2	0.157506	0.068750	0.931250	0.798249	0.209302	0.790698	00:40	3	0.155925	0.056250	0.943750	0.527339	0.162791	0.837209	00:40	4	0.137334	0.043750	0.956250	0.561539	0.139535	0.860465	00:39																										
epoch	train_loss	train_error_rate	train_accuracy	valid_loss	valid_error_rate	valid_accuracy	time																																																																					
0	0.136299	0.037500	0.962500	0.512523	0.162791	0.837209	00:39																																																																					
1	0.110495	0.031250	0.968750	0.626146	0.209302	0.790698	00:39																																																																					
2	0.157506	0.068750	0.931250	0.798249	0.209302	0.790698	00:40																																																																					
3	0.155925	0.056250	0.943750	0.527339	0.162791	0.837209	00:40																																																																					
4	0.137334	0.043750	0.956250	0.561539	0.139535	0.860465	00:39																																																																					
Avaliação do Fine tuning	Matriz de confusão	<p>Análise de requisitos Preparação de dados Transfer learning Avaliação transfer learning Fine tuning (opcional) Avaliação fine tuning (opcional)</p> <p>Matriz de confusão ?</p> <p>Ver matriz de confusão</p> <table border="1"> <thead> <tr> <th colspan="2"></th> <th colspan="5">Actual</th> </tr> <tr> <th colspan="2"></th> <th>Azeira</th> <th>Capororoca</th> <th>Embauba</th> <th>Jeriva</th> <th>Mulungu</th> <th>Pitangueira</th> </tr> </thead> <tbody> <tr> <th rowspan="5">Actual</th> <th>Azeira</th> <td>4</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <th>Capororoca</th> <td>0</td> <td>7</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <th>Embauba</th> <td>0</td> <td>0</td> <td>4</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>Jeriva</th> <td>0</td> <td>0</td> <td>0</td> <td>4</td> <td>0</td> <td>0</td> </tr> <tr> <th>Mulungu</th> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>4</td> <td>2</td> </tr> <tr> <th>Pitangueira</th> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>7</td> </tr> <tr> <th colspan="2"></th> <th colspan="5">Predicted</th> </tr> <tr> <th colspan="2"></th> <th>Azeira</th> <th>Capororoca</th> <th>Embauba</th> <th>Jeriva</th> <th>Mulungu</th> <th>Pitangueira</th> </tr> </tbody> </table>			Actual							Azeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira	Actual	Azeira	4	0	1	0	0	1	Capororoca	0	7	0	0	0	1	Embauba	0	0	4	0	0	0	Jeriva	0	0	0	4	0	0	Mulungu	0	0	1	0	4	2	Pitangueira	0	0	0	0	0	1	7			Predicted							Azeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira
		Actual																																																																										
		Azeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira																																																																					
Actual	Azeira	4	0	1	0	0	1																																																																					
	Capororoca	0	7	0	0	0	1																																																																					
	Embauba	0	0	4	0	0	0																																																																					
	Jeriva	0	0	0	4	0	0																																																																					
	Mulungu	0	0	1	0	4	2																																																																					
Pitangueira	0	0	0	0	0	1	7																																																																					
		Predicted																																																																										
		Azeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira																																																																					

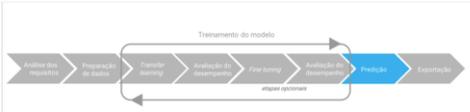
	Acurácia por categoria	<p>Métricas por categoria ?</p> <table border="1"> <thead> <tr> <th colspan="5">Ver métricas por categoria</th> </tr> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Aroeira</td> <td>1.00</td> <td>0.67</td> <td>0.80</td> <td>6</td> </tr> <tr> <td>Capororoca</td> <td>1.00</td> <td>0.88</td> <td>0.93</td> <td>8</td> </tr> <tr> <td>Embauba</td> <td>0.80</td> <td>1.00</td> <td>0.89</td> <td>8</td> </tr> <tr> <td>Jeriva</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>4</td> </tr> <tr> <td>Mulungu</td> <td>0.86</td> <td>0.67</td> <td>0.75</td> <td>9</td> </tr> <tr> <td>Pitangueira</td> <td>0.64</td> <td>0.88</td> <td>0.74</td> <td>8</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.84</td> <td>43</td> </tr> <tr> <td>macro avg</td> <td>0.88</td> <td>0.85</td> <td>0.85</td> <td>43</td> </tr> <tr> <td>weighted avg</td> <td>0.87</td> <td>0.84</td> <td>0.84</td> <td>43</td> </tr> </tbody> </table>	Ver métricas por categoria						precision	recall	f1-score	support	Aroeira	1.00	0.67	0.80	6	Capororoca	1.00	0.88	0.93	8	Embauba	0.80	1.00	0.89	8	Jeriva	1.00	1.00	1.00	4	Mulungu	0.86	0.67	0.75	9	Pitangueira	0.64	0.88	0.74	8	accuracy			0.84	43	macro avg	0.88	0.85	0.85	43	weighted avg	0.87	0.84	0.84	43
Ver métricas por categoria																																																									
	precision	recall	f1-score	support																																																					
Aroeira	1.00	0.67	0.80	6																																																					
Capororoca	1.00	0.88	0.93	8																																																					
Embauba	0.80	1.00	0.89	8																																																					
Jeriva	1.00	1.00	1.00	4																																																					
Mulungu	0.86	0.67	0.75	9																																																					
Pitangueira	0.64	0.88	0.74	8																																																					
accuracy			0.84	43																																																					
macro avg	0.88	0.85	0.85	43																																																					
weighted avg	0.87	0.84	0.84	43																																																					
	Principais perdas	<p>Principais perdas ?</p> <p>Qtd. <input type="range" value="3"/> 3</p> <p>Ver principais perdas</p> <p>Prediction/Actual/Loss/Probability</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Embauba/Mulungu / 8.16 / 0.52</p>  </div> <div style="text-align: center;"> <p>Pitangueira/Capororoca / 5.12 / 0.69</p>  </div> <div style="text-align: center;"> <p>Embauba/Aroeira / 1.99 / 0.84</p>  </div> </div>																																																							
	Mais Confundidos	<p>Mais confundidos ?</p> <p>vez(es) confundido <input type="range" value="2"/> 2</p> <p>Ver mais confundidos</p> <p>Mulungu foi confundida com Pitangueira e esta combinação em particular ocorreu 2 vez(es)</p>																																																							
Predição	<i>Upload</i>	<p>Análise de requisitos Preparação de dados Transfer learning Avaliação transfer learning Fine tuning (opcional) Avaliação fine tuning (opcional) Predição</p> <p>Tratamento do modelo</p>  <p>Predição com upload ?</p> <p>Upload (1)</p> <p>Predição</p> <p>Aguarde, realizando predição...</p>  <p>Isto é um(a) Embauba, com a probabilidade de 0.000002</p>																																																							
	Predição																																																								
Exportação	Exportar modelo PKL	<p>Análise de requisitos Preparação de dados Transfer learning Avaliação transfer learning Fine tuning (opcional) Avaliação fine tuning (opcional) Predição Exportação</p> <p>Tratamento do modelo</p>  <p>Exportação do modelo ?</p> <p>Export/</p> <p>Exportar modelo PKL</p> <p>Exportar modelo ONNX</p> <p>Exportado para: "contentigativeMy Drive\Export\export_model.pkl com sucesso</p> <p>Exportado para: "contentigativeMy Drive\Export\export_model_onnx.onnx com sucesso</p>																																																							
	Exportar modelo ONNX																																																								

Tabela 15 - Resultado obtido na interface visual

4.3.5. Distribuição

Todo o código desenvolvido está disponível no repositório de códigos da UFSC⁴ sob a licença BSD 3-clause.

O pacote *vulcan-colab* também estará disponível na biblioteca PyPI para ser importado diretamente em qualquer *notebook* do Colab. Um notebook de exemplo também estará disponível. Todos os links, instruções e ajuda se encontram na documentação do Vulcan⁵.

```

Vulcan - PIP.ipynb
Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alteraç...
+ Código + Texto
[1] 1 ! [ -e /content ] && pip install -Uqq fastai
    2 !pip install -i https://pypi.org/simple/ vulcan_colab
    186 kB 7.7 MB/s
    56 kB 5.8 MB/s
    Looking in indexes: https://pypi.org/simple/
    Collecting vulcan_colab
    Downloading https://files.pythonhosted.org/packages/60/c9/82e4f8cc9794c12ef77f0a0bdc5912f311c5311c5
    1.9 MB 7.1 MB/s
    Building wheels for collected packages: vulcan-colab
    Building wheel for vulcan-colab (setup.py) ... done
    Created wheel for vulcan-colab: filename=vulcan_colab-0.1.1-py3-none-any.whl size=1853325 sha256=77b
    Stored in directory: /root/.cache/pip/wheels/7f/83/c9/551e80eee93e9df4708603ab8289812da333a1980f26e2
    Successfully built vulcan-colab
    Installing collected packages: vulcan-colab
    Successfully installed vulcan-colab-0.1.1

[2] 1 from visual_layer import vulcan_colab
    Fim da primeira execução, exectue vulcan_colab.visual_layer() em outra célula de código

1 vulcan_colab.visual_layer()
Mounted at /content/gdrive
Google Drive conectado com sucesso

COMPUTAÇÃO NA ESCOLA
INCoD ine UNIVERSIDADE FEDERAL DE SANTA CATARINA

Análise de requisitos Preparação de dados Transfer learning Avaliação transfer learning Fine tuning (opcional)
Avaliação fine tuning (opcional) Predição Exportação

Treinamento do modelo
Análise de dados Preparação de dados Treinamento Avaliação Avaliação final
  
```

Figura 11 - Exemplo da instalação e execução do Vulcan

⁴ <https://codigos.ufsc.br/gqs/colabvisualallayer> *Biblioteca ainda sendo registrada pela UFSC no INPI

⁵ <https://vulcan-colab.readthedocs.io/>

4.4. Exemplo de Aplicação: Modelo de ML de reconhecimento de árvores nativas de Santa Catarina

A fim de validar o desenvolvimento do Visual layer classification foi realizada uma aplicação do Vulcan a um exemplo real. Em cooperação com pesquisadores do departamento de Biologia da UFSC e com base no projeto Inventário florístico da UFSC (UFSC SUSTENTÁVEL, 2016) foram coletadas e rotuladas imagens referentes a 6 espécies nativas de árvores conforme a Tabela 16. Também existem outras pesquisas dos membros da iniciativa Computação na Escola/INCoD/INE/UFSC com o objetivo de criar uma unidade instrucional para esta atividade assim como para a avaliação.

Nome popular	Espécie	Qtd de árvores no Campus
Aroeira-Vermelha	Schinus terebinthifolius	155
Jerivá	Syagrus romanzoffiana	150
Mulungu, Eritrina	Erythrina speciosa	48
Capororoca	Myrsine guianensis	47
Embaúba	Cecropia glaziovii	43
Pitangueira	Eugenia uniflora	39

Tabela 16 - Árvores nativas selecionadas para a criação do modelo.

4.4.1. Conjunto de dados

Como resultado da coleta e rotulação das imagens das árvores em 2020 foi criado um conjunto de dados totalizando 214 imagens. Mais informações sobre o conjunto de dados resultante são mostrados na Tabela 17.

Conjunto de dados	
Descrição dos dados	Conjunto de imagens ⁶ de 6 espécies nativas entre as mais frequentes no campus da UFSC
Origem dos dados (coleta própria/uso de conjunto de dados pré-existente, p.ex. do kaggle)	Coleta própria realizada pelos integrantes do laboratório em 2020
Quantidade total de dados	214

⁶ https://ufscbr-my.sharepoint.com/:f/g/personal/jean_hauck_ufsc_br/EkpANa4nooZIoJPmeNVRhWYB_BQJYawcxcDXJgNCsFWQ4g?e=4Tj64x

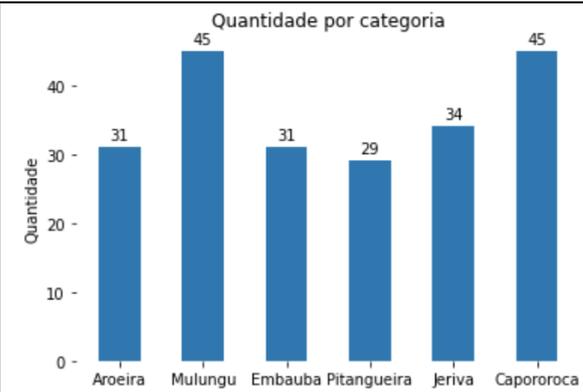
Distribuição dos dados por categoria	
Labeling	Realizada em cooperação com o departamento de Biologia da UFSC e com base no projeto Inventário florístico da UFSC (UFSC SUSTENTÁVEL, 2016)
Tamanho de imagens	224x224 pixels
Tamanho do batch	16
Dataset splitting	80% para treinamento (171 imagens) 20% para validação (43 imagens)

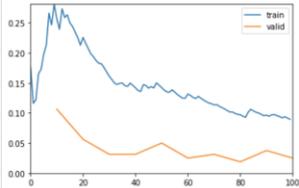
Tabela 17 - Informações do conjunto de dados

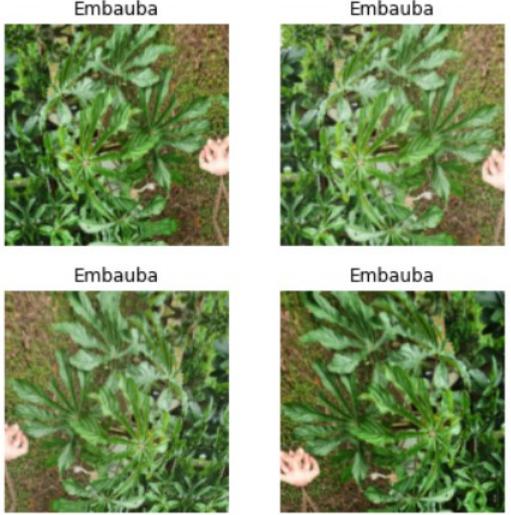
4.4.2. Treinamento e desempenho

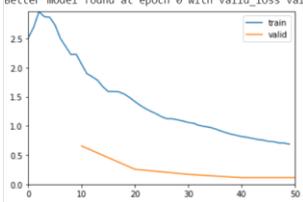
Foram criados e treinados 3 modelos, cada um com diferentes arquiteturas e hiperparâmetros. A análise de requisitos foi a mesma para todos os modelos e é mostrada na Tabela 18. Na Tabela 19 temos os detalhes do treinamento e avaliação de cada modelo.

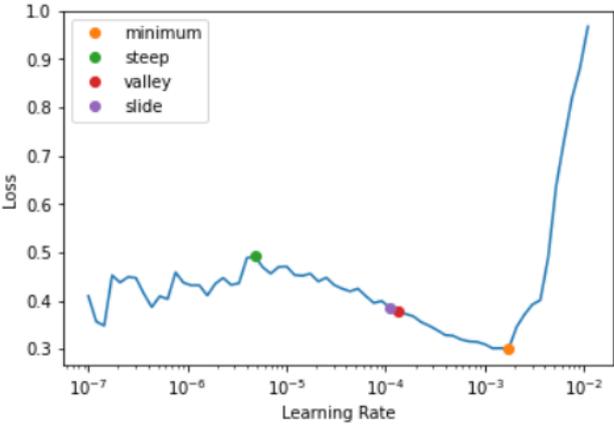
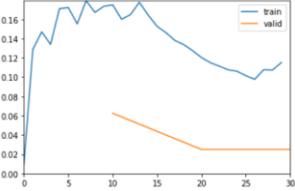
Análise de requisitos	Tipo da tarefa	Single-label classificação de imagens
	Categorias	categorias de espécies de árvores nativas/endêmicas de SC/Brasil (Aroeira-vermelha, Capororoca, Embaúba, Jerivá, Mulungu, Pitangueira Experiência)
	Fonte de dados	Conjunto de dados de árvores
	Quantidade de dados	215 no total, aproximadamente 30 por categoria
	Padronização das imagens	224x224 pixels
	Rotulação dos dados	Padrão do conjunto de dados
	Desempenho	Acurácia (total/por categoria): No mínimo 0.75, F1 score: No mínimo 0.75.

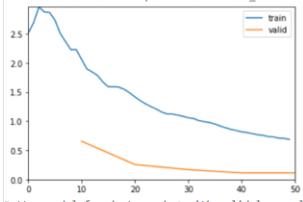
Tabela 18 - Análise de requisitos para o treinamento do modelo.

Modelo	Processo de ML	Tarefa	Parâmetro/Resultado																																																																																								
Modelo 1		Padronização das imagens: Redimensionamento	Padrão (224x224 pixels)																																																																																								
		Padronização das imagens: Transformações e aumentações	Padrão = Desabilitado																																																																																								
		Separação do conjunto de dados de treinamento e validação	Padrão = 20%																																																																																								
		Tamanho do Batch	Padrão = 16																																																																																								
	Transfer Learning	Seleção da arquitectura	resnet18																																																																																								
		Métricas do acompanhamento do treinamento	Taxa de erro e acurácia																																																																																								
		Epochs	10																																																																																								
		Treinar modelo	<table border="1"> <thead> <tr> <th>epoch</th> <th>train_loss</th> <th>train_error_rate</th> <th>train_accuracy</th> <th>valid_loss</th> <th>valid_error_rate</th> <th>valid_accuracy</th> <th>time</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.280980</td><td>0.106250</td><td>0.893750</td><td>0.256481</td><td>0.095238</td><td>0.904762</td><td>00:40</td></tr> <tr><td>1</td><td>0.212776</td><td>0.056250</td><td>0.943750</td><td>0.227947</td><td>0.071429</td><td>0.928571</td><td>00:39</td></tr> <tr><td>2</td><td>0.168160</td><td>0.031250</td><td>0.968750</td><td>0.336962</td><td>0.119048</td><td>0.880952</td><td>00:39</td></tr> <tr><td>3</td><td>0.145879</td><td>0.031250</td><td>0.968750</td><td>0.492469</td><td>0.142857</td><td>0.857143</td><td>00:39</td></tr> <tr><td>4</td><td>0.146272</td><td>0.050000</td><td>0.950000</td><td>0.495165</td><td>0.095238</td><td>0.904762</td><td>00:39</td></tr> <tr><td>5</td><td>0.124546</td><td>0.025000</td><td>0.975000</td><td>0.367252</td><td>0.047619</td><td>0.952381</td><td>00:40</td></tr> <tr><td>6</td><td>0.115522</td><td>0.031250</td><td>0.968750</td><td>0.285698</td><td>0.047619</td><td>0.952381</td><td>00:39</td></tr> <tr><td>7</td><td>0.097634</td><td>0.018750</td><td>0.981250</td><td>0.216737</td><td>0.071429</td><td>0.928571</td><td>00:39</td></tr> <tr><td>8</td><td>0.095481</td><td>0.037500</td><td>0.962500</td><td>0.204727</td><td>0.047619</td><td>0.952381</td><td>00:39</td></tr> <tr><td>9</td><td>0.089523</td><td>0.025000</td><td>0.975000</td><td>0.205013</td><td>0.047619</td><td>0.952381</td><td>00:39</td></tr> </tbody> </table> <p>Better model found at epoch 0 with valid_loss value: 0.2564808138264282. Better model found at epoch 0 with valid_loss value: 0.2564808138264282.  Better model found at epoch 1 with valid_loss value: 0.2279466837644577. Better model found at epoch 1 with valid_loss value: 0.2279466837644577. Better model found at epoch 7 with valid_loss value: 0.21673688292583357. Better model found at epoch 7 with valid_loss value: 0.21673688292583357. Better model found at epoch 8 with valid_loss value: 0.2047274261713828. Better model found at epoch 8 with valid_loss value: 0.2047274261713828.</p>	epoch	train_loss	train_error_rate	train_accuracy	valid_loss	valid_error_rate	valid_accuracy	time	0	0.280980	0.106250	0.893750	0.256481	0.095238	0.904762	00:40	1	0.212776	0.056250	0.943750	0.227947	0.071429	0.928571	00:39	2	0.168160	0.031250	0.968750	0.336962	0.119048	0.880952	00:39	3	0.145879	0.031250	0.968750	0.492469	0.142857	0.857143	00:39	4	0.146272	0.050000	0.950000	0.495165	0.095238	0.904762	00:39	5	0.124546	0.025000	0.975000	0.367252	0.047619	0.952381	00:40	6	0.115522	0.031250	0.968750	0.285698	0.047619	0.952381	00:39	7	0.097634	0.018750	0.981250	0.216737	0.071429	0.928571	00:39	8	0.095481	0.037500	0.962500	0.204727	0.047619	0.952381	00:39	9	0.089523	0.025000	0.975000	0.205013	0.047619	0.952381	00:39
		epoch	train_loss	train_error_rate	train_accuracy	valid_loss	valid_error_rate	valid_accuracy	time																																																																																		
		0	0.280980	0.106250	0.893750	0.256481	0.095238	0.904762	00:40																																																																																		
1	0.212776	0.056250	0.943750	0.227947	0.071429	0.928571	00:39																																																																																				
2	0.168160	0.031250	0.968750	0.336962	0.119048	0.880952	00:39																																																																																				
3	0.145879	0.031250	0.968750	0.492469	0.142857	0.857143	00:39																																																																																				
4	0.146272	0.050000	0.950000	0.495165	0.095238	0.904762	00:39																																																																																				
5	0.124546	0.025000	0.975000	0.367252	0.047619	0.952381	00:40																																																																																				
6	0.115522	0.031250	0.968750	0.285698	0.047619	0.952381	00:39																																																																																				
7	0.097634	0.018750	0.981250	0.216737	0.071429	0.928571	00:39																																																																																				
8	0.095481	0.037500	0.962500	0.204727	0.047619	0.952381	00:39																																																																																				
9	0.089523	0.025000	0.975000	0.205013	0.047619	0.952381	00:39																																																																																				
Avaliação do Transfer Learning	Matriz de confusão	Confusion matrix	<table border="1"> <thead> <tr> <th></th> <th>Aroeira</th> <th>Capororoca</th> <th>Embauba</th> <th>Jeriva</th> <th>Mulungu</th> <th>Pitangueira</th> </tr> </thead> <tbody> <tr> <th>Aroeira</th> <td>7</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>Capororoca</th> <td>0</td> <td>6</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>Embauba</th> <td>0</td> <td>1</td> <td>5</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>Jeriva</th> <td>0</td> <td>0</td> <td>0</td> <td>7</td> <td>0</td> <td>0</td> </tr> <tr> <th>Mulungu</th> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>9</td> <td>0</td> </tr> <tr> <th>Pitangueira</th> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>6</td> </tr> </tbody> </table>		Aroeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira	Aroeira	7	0	0	0	0	0	Capororoca	0	6	0	0	0	0	Embauba	0	1	5	0	0	0	Jeriva	0	0	0	7	0	0	Mulungu	0	1	0	0	9	0	Pitangueira	0	0	0	0	0	6																																							
			Aroeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira																																																																																			
Aroeira	7	0	0	0	0	0																																																																																					
Capororoca	0	6	0	0	0	0																																																																																					
Embauba	0	1	5	0	0	0																																																																																					
Jeriva	0	0	0	7	0	0																																																																																					
Mulungu	0	1	0	0	9	0																																																																																					
Pitangueira	0	0	0	0	0	6																																																																																					

		Acurácia por categoria	<table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Aroeira</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>7</td> </tr> <tr> <td>Capororoca</td> <td>0.75</td> <td>1.00</td> <td>0.86</td> <td>6</td> </tr> <tr> <td>Embauba</td> <td>1.00</td> <td>0.83</td> <td>0.91</td> <td>6</td> </tr> <tr> <td>Jeriva</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>7</td> </tr> <tr> <td>Mulungu</td> <td>1.00</td> <td>0.90</td> <td>0.95</td> <td>10</td> </tr> <tr> <td>Pitangueira</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>6</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.95</td> <td>42</td> </tr> <tr> <td>macro avg</td> <td>0.96</td> <td>0.96</td> <td>0.95</td> <td>42</td> </tr> <tr> <td>weighted avg</td> <td>0.96</td> <td>0.95</td> <td>0.95</td> <td>42</td> </tr> </tbody> </table>		precision	recall	f1-score	support	Aroeira	1.00	1.00	1.00	7	Capororoca	0.75	1.00	0.86	6	Embauba	1.00	0.83	0.91	6	Jeriva	1.00	1.00	1.00	7	Mulungu	1.00	0.90	0.95	10	Pitangueira	1.00	1.00	1.00	6	accuracy			0.95	42	macro avg	0.96	0.96	0.95	42	weighted avg	0.96	0.95	0.95	42
	precision	recall	f1-score	support																																																	
Aroeira	1.00	1.00	1.00	7																																																	
Capororoca	0.75	1.00	0.86	6																																																	
Embauba	1.00	0.83	0.91	6																																																	
Jeriva	1.00	1.00	1.00	7																																																	
Mulungu	1.00	0.90	0.95	10																																																	
Pitangueira	1.00	1.00	1.00	6																																																	
accuracy			0.95	42																																																	
macro avg	0.96	0.96	0.95	42																																																	
weighted avg	0.96	0.95	0.95	42																																																	
	Tempo	Tempo total de uso	8 minutos																																																		
Modelo 2		Padronização das imagens: Redimensionamento	Padrão (224x224 pixels)																																																		
		Padronização das imagens: Transformações e aumentações	<p>Transformações e Aumentações 2</p> <p><input type="checkbox"/> Desabilitar</p> <p>Zoom máximo <input type="range"/> 1.10</p> <p>Zoom mínimo <input type="range"/> 0.90</p> <p>Espehar horizontalmente <input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não</p> <p>Espehar verticalmente <input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não</p> <p>% de intensidade <input type="range"/> 1.00</p> <p>Rotação máxima <input type="range"/> 10</p> <p>Iluminação máxima <input type="range"/> 0.20</p> <p>Intensidade iluminação <input type="range"/> 0.75</p> <p>Inclinação <input type="range"/> 0.20</p>																																																		
		Separação do conjunto de dados de treinamento e validação	Padrão = 20%																																																		
		Tamanho do Batch	Padrão = 16																																																		
		Resultado das aumentações																																																			

	Transfer Learning	Seleção da arquitectura	vgg_16bn																																																		
		Métricas do acompanhamento do treinamento	Taxa de erro e acurácia																																																		
		Epochs	5																																																		
		Treinar modelo	<pre> epoch train_loss train_error_rate train_accuracy valid_loss valid_error_rate valid_accuracy time 0 2.229062 0.656250 0.343750 1.146644 0.418605 0.581395 02:22 1 1.487249 0.256250 0.743750 0.859257 0.209302 0.790698 02:10 2 1.085075 0.168750 0.831250 0.858769 0.302326 0.697674 02:12 3 0.840011 0.112500 0.887500 0.883054 0.232558 0.767442 02:11 4 0.686912 0.112500 0.887500 0.881452 0.186046 0.813953 02:11 </pre> <p>Better model found at epoch 0 with valid_loss value: 1.146644115447998. Better model found at epoch 0 with valid_loss value: 1.146644115447998.</p>  <p>Better model found at epoch 1 with valid_loss value: 0.8592568039894104. Better model found at epoch 1 with valid_loss value: 0.8592568039894104. Better model found at epoch 2 with valid_loss value: 0.858769416809082. Better model found at epoch 2 with valid_loss value: 0.858769416809082.</p>																																																		
Avaliação do Transfer Learning	Matriz de confusão	<p>Confusion matrix</p> <table border="1"> <thead> <tr> <th></th> <th>Aroeira</th> <th>Capororoca</th> <th>Embauba</th> <th>Jeriva</th> <th>Mulungu</th> <th>Pitangueira</th> </tr> </thead> <tbody> <tr> <th>Aroeira</th> <td>2</td> <td>1</td> <td>3</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <th>Capororoca</th> <td>0</td> <td>7</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>Embauba</th> <td>0</td> <td>0</td> <td>5</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <th>Jeriva</th> <td>0</td> <td>0</td> <td>0</td> <td>4</td> <td>0</td> <td>0</td> </tr> <tr> <th>Mulungu</th> <td>0</td> <td>1</td> <td>2</td> <td>0</td> <td>8</td> <td>0</td> </tr> <tr> <th>Pitangueira</th> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>4</td> </tr> </tbody> </table>		Aroeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira	Aroeira	2	1	3	1	0	1	Capororoca	0	7	1	0	0	0	Embauba	0	0	5	1	0	0	Jeriva	0	0	0	4	0	0	Mulungu	0	1	2	0	8	0	Pitangueira	0	0	1	1	0	4		
		Aroeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira																																														
Aroeira	2	1	3	1	0	1																																															
Capororoca	0	7	1	0	0	0																																															
Embauba	0	0	5	1	0	0																																															
Jeriva	0	0	0	4	0	0																																															
Mulungu	0	1	2	0	8	0																																															
Pitangueira	0	0	1	1	0	4																																															
		Acurácia por categoria	<table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Aroeira</td> <td>1.00</td> <td>0.25</td> <td>0.40</td> <td>8</td> </tr> <tr> <td>Capororoca</td> <td>0.78</td> <td>0.88</td> <td>0.82</td> <td>8</td> </tr> <tr> <td>Embauba</td> <td>0.42</td> <td>0.83</td> <td>0.56</td> <td>6</td> </tr> <tr> <td>Jeriva</td> <td>0.57</td> <td>1.00</td> <td>0.73</td> <td>4</td> </tr> <tr> <td>Mulungu</td> <td>1.00</td> <td>0.73</td> <td>0.84</td> <td>11</td> </tr> <tr> <td>Pitangueira</td> <td>0.80</td> <td>0.67</td> <td>0.73</td> <td>6</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.70</td> <td>43</td> </tr> <tr> <td>macro avg</td> <td>0.76</td> <td>0.73</td> <td>0.68</td> <td>43</td> </tr> <tr> <td>weighted avg</td> <td>0.81</td> <td>0.70</td> <td>0.69</td> <td>43</td> </tr> </tbody> </table>		precision	recall	f1-score	support	Aroeira	1.00	0.25	0.40	8	Capororoca	0.78	0.88	0.82	8	Embauba	0.42	0.83	0.56	6	Jeriva	0.57	1.00	0.73	4	Mulungu	1.00	0.73	0.84	11	Pitangueira	0.80	0.67	0.73	6	accuracy			0.70	43	macro avg	0.76	0.73	0.68	43	weighted avg	0.81	0.70	0.69	43
	precision	recall	f1-score	support																																																	
Aroeira	1.00	0.25	0.40	8																																																	
Capororoca	0.78	0.88	0.82	8																																																	
Embauba	0.42	0.83	0.56	6																																																	
Jeriva	0.57	1.00	0.73	4																																																	
Mulungu	1.00	0.73	0.84	11																																																	
Pitangueira	0.80	0.67	0.73	6																																																	
accuracy			0.70	43																																																	
macro avg	0.76	0.73	0.68	43																																																	
weighted avg	0.81	0.70	0.69	43																																																	

		<p>Encontrar melhor taxa</p>	 <p> <code>lr_min: 0.0001737800776027143</code> <code>lr_steep: 4.786300905834651e-06</code> <code>lr_valley: 0.00013182566908653826</code> <code>lr_slide: 0.00010964782268274575</code> </p>																																																	
		<p>Epochs fine tuning</p>	<p>3</p>																																																	
		<p>Treinar modelo otimizado</p>	<table border="1"> <thead> <tr> <th>epoch</th> <th>train_loss</th> <th>train_error_rate</th> <th>train_accuracy</th> <th>valid_loss</th> <th>valid_error_rate</th> <th>valid_accuracy</th> <th>time</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0.173503</td> <td>0.062500</td> <td>0.937500</td> <td>0.619776</td> <td>0.186046</td> <td>0.813953</td> <td>02:14</td> </tr> <tr> <td>1</td> <td>0.127475</td> <td>0.025000</td> <td>0.975000</td> <td>0.625603</td> <td>0.209302</td> <td>0.790698</td> <td>02:13</td> </tr> <tr> <td>2</td> <td>0.115114</td> <td>0.025000</td> <td>0.975000</td> <td>0.621372</td> <td>0.162791</td> <td>0.837209</td> <td>02:14</td> </tr> </tbody> </table> <p> Better model found at epoch 0 with valid_loss value: 0.6197757720947266. Better model found at epoch 0 with valid_loss value: 0.6197757720947266. </p> 	epoch	train_loss	train_error_rate	train_accuracy	valid_loss	valid_error_rate	valid_accuracy	time	0	0.173503	0.062500	0.937500	0.619776	0.186046	0.813953	02:14	1	0.127475	0.025000	0.975000	0.625603	0.209302	0.790698	02:13	2	0.115114	0.025000	0.975000	0.621372	0.162791	0.837209	02:14																	
epoch	train_loss	train_error_rate	train_accuracy	valid_loss	valid_error_rate	valid_accuracy	time																																													
0	0.173503	0.062500	0.937500	0.619776	0.186046	0.813953	02:14																																													
1	0.127475	0.025000	0.975000	0.625603	0.209302	0.790698	02:13																																													
2	0.115114	0.025000	0.975000	0.621372	0.162791	0.837209	02:14																																													
<p>Avaliação do Fine tuning</p>		<p>Matriz de confusão</p>	<p>Confusion matrix</p> <table border="1"> <thead> <tr> <th>Actual \ Predicted</th> <th>Aroeira</th> <th>Capororoca</th> <th>Embauba</th> <th>Jeriva</th> <th>Mulungu</th> <th>Pitangueira</th> </tr> </thead> <tbody> <tr> <th>Aroeira</th> <td>4</td> <td>1</td> <td>2</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <th>Capororoca</th> <td>0</td> <td>7</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <th>Embauba</th> <td>0</td> <td>0</td> <td>6</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <th>Jeriva</th> <td>0</td> <td>0</td> <td>0</td> <td>4</td> <td>0</td> <td>0</td> </tr> <tr> <th>Mulungu</th> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>10</td> <td>1</td> </tr> <tr> <th>Pitangueira</th> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>4</td> </tr> </tbody> </table>	Actual \ Predicted	Aroeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira	Aroeira	4	1	2	1	0	0	Capororoca	0	7	0	1	0	0	Embauba	0	0	6	0	0	0	Jeriva	0	0	0	4	0	0	Mulungu	0	0	0	0	10	1	Pitangueira	0	0	1	1	0	4
		Actual \ Predicted	Aroeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira																																												
Aroeira	4	1	2	1	0	0																																														
Capororoca	0	7	0	1	0	0																																														
Embauba	0	0	6	0	0	0																																														
Jeriva	0	0	0	4	0	0																																														
Mulungu	0	0	0	0	10	1																																														
Pitangueira	0	0	1	1	0	4																																														
<p>Acurácia por categoria</p>	<table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Aroeira</td> <td>1.00</td> <td>0.50</td> <td>0.67</td> <td>8</td> </tr> <tr> <td>Capororoca</td> <td>0.88</td> <td>0.88</td> <td>0.88</td> <td>8</td> </tr> <tr> <td>Embauba</td> <td>0.67</td> <td>1.00</td> <td>0.80</td> <td>6</td> </tr> <tr> <td>Jeriva</td> <td>0.57</td> <td>1.00</td> <td>0.73</td> <td>4</td> </tr> <tr> <td>Mulungu</td> <td>1.00</td> <td>0.91</td> <td>0.95</td> <td>11</td> </tr> <tr> <td>Pitangueira</td> <td>0.80</td> <td>0.67</td> <td>0.73</td> <td>6</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.81</td> <td>43</td> </tr> <tr> <td>macro avg</td> <td>0.82</td> <td>0.83</td> <td>0.79</td> <td>43</td> </tr> <tr> <td>weighted avg</td> <td>0.86</td> <td>0.81</td> <td>0.81</td> <td>43</td> </tr> </tbody> </table>		precision	recall	f1-score	support	Aroeira	1.00	0.50	0.67	8	Capororoca	0.88	0.88	0.88	8	Embauba	0.67	1.00	0.80	6	Jeriva	0.57	1.00	0.73	4	Mulungu	1.00	0.91	0.95	11	Pitangueira	0.80	0.67	0.73	6	accuracy			0.81	43	macro avg	0.82	0.83	0.79	43	weighted avg	0.86	0.81	0.81	43	
	precision	recall	f1-score	support																																																
Aroeira	1.00	0.50	0.67	8																																																
Capororoca	0.88	0.88	0.88	8																																																
Embauba	0.67	1.00	0.80	6																																																
Jeriva	0.57	1.00	0.73	4																																																
Mulungu	1.00	0.91	0.95	11																																																
Pitangueira	0.80	0.67	0.73	6																																																
accuracy			0.81	43																																																
macro avg	0.82	0.83	0.79	43																																																
weighted avg	0.86	0.81	0.81	43																																																

	Tempo	Tempo total de uso	20 minutos																																																
Modelo 3		Padronização das imagens: Redimensionamento	Padrão (224x224 pixels)																																																
		Padronização das imagens: Transformações e aumentações	Padrão = Desabilitado																																																
		Separação do conjunto de dados de treinamento e validação	Padrão = 20%																																																
		Tamanho do Batch	16																																																
	Transfer Learning	Seleção da arquitectura	densenet121																																																
		Métricas do acompanhamento do treinamento	Taxa de erro e acurácia																																																
		Epochs	20																																																
		Treinar modelo	<table border="1"> <thead> <tr> <th>epoch</th> <th>train_loss</th> <th>train_error_rate</th> <th>train_accuracy</th> <th>valid_loss</th> <th>valid_error_rate</th> <th>valid_accuracy</th> <th>time</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2.229062</td> <td>0.656250</td> <td>0.343750</td> <td>1.146644</td> <td>0.418605</td> <td>0.581395</td> <td>02:22</td> </tr> <tr> <td>1</td> <td>1.487249</td> <td>0.256250</td> <td>0.743750</td> <td>0.859257</td> <td>0.209302</td> <td>0.790698</td> <td>02:10</td> </tr> <tr> <td>2</td> <td>1.085075</td> <td>0.168750</td> <td>0.831250</td> <td>0.858769</td> <td>0.302326</td> <td>0.697674</td> <td>02:12</td> </tr> <tr> <td>3</td> <td>0.840011</td> <td>0.112500</td> <td>0.887500</td> <td>0.883054</td> <td>0.232558</td> <td>0.767442</td> <td>02:11</td> </tr> <tr> <td>4</td> <td>0.686912</td> <td>0.112500</td> <td>0.887500</td> <td>0.881452</td> <td>0.186046</td> <td>0.813953</td> <td>02:11</td> </tr> </tbody> </table> <p>Better model found at epoch 0 with valid_loss value: 1.146644115447998. Better model found at epoch 0 with valid_loss value: 1.146644115447998.</p>  <p>Better model found at epoch 1 with valid_loss value: 0.8592568039894104. Better model found at epoch 1 with valid_loss value: 0.8592568039894104. Better model found at epoch 2 with valid_loss value: 0.858769416809082. Better model found at epoch 2 with valid_loss value: 0.858769416809082.</p>	epoch	train_loss	train_error_rate	train_accuracy	valid_loss	valid_error_rate	valid_accuracy	time	0	2.229062	0.656250	0.343750	1.146644	0.418605	0.581395	02:22	1	1.487249	0.256250	0.743750	0.859257	0.209302	0.790698	02:10	2	1.085075	0.168750	0.831250	0.858769	0.302326	0.697674	02:12	3	0.840011	0.112500	0.887500	0.883054	0.232558	0.767442	02:11	4	0.686912	0.112500	0.887500	0.881452	0.186046	0.813953	02:11
		epoch	train_loss	train_error_rate	train_accuracy	valid_loss	valid_error_rate	valid_accuracy	time																																										
		0	2.229062	0.656250	0.343750	1.146644	0.418605	0.581395	02:22																																										
1	1.487249	0.256250	0.743750	0.859257	0.209302	0.790698	02:10																																												
2	1.085075	0.168750	0.831250	0.858769	0.302326	0.697674	02:12																																												
3	0.840011	0.112500	0.887500	0.883054	0.232558	0.767442	02:11																																												
4	0.686912	0.112500	0.887500	0.881452	0.186046	0.813953	02:11																																												

Avaliação do Transfer Learning	Matriz de confusão	<p style="text-align: center;">Confusion matrix</p> <table border="1"> <tr> <td>Aroeira</td> <td>7</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Capororoca</td> <td>1</td> <td>6</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Embauba</td> <td>0</td> <td>0</td> <td>6</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Jeriva</td> <td>0</td> <td>0</td> <td>0</td> <td>4</td> <td>0</td> <td>0</td> </tr> <tr> <td>Mulungu</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>10</td> <td>1</td> </tr> <tr> <td>Pitangueira</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>4</td> </tr> <tr> <td></td> <td>Aroeira</td> <td>Capororoca</td> <td>Embauba</td> <td>Jeriva</td> <td>Mulungu</td> <td>Pitangueira</td> </tr> </table> <p style="text-align: center;">Predicted</p>	Aroeira	7	0	1	0	0	0	Capororoca	1	6	1	0	0	0	Embauba	0	0	6	0	0	0	Jeriva	0	0	0	4	0	0	Mulungu	0	0	0	0	10	1	Pitangueira	1	0	0	0	1	4		Aroeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira
	Aroeira	7	0	1	0	0	0																																												
	Capororoca	1	6	1	0	0	0																																												
Embauba	0	0	6	0	0	0																																													
Jeriva	0	0	0	4	0	0																																													
Mulungu	0	0	0	0	10	1																																													
Pitangueira	1	0	0	0	1	4																																													
	Aroeira	Capororoca	Embauba	Jeriva	Mulungu	Pitangueira																																													
Acurácia por categoria	<table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>Aroeira</td> <td>0.78</td> <td>0.88</td> <td>0.82</td> <td>8</td> </tr> <tr> <td>Capororoca</td> <td>1.00</td> <td>0.75</td> <td>0.86</td> <td>8</td> </tr> <tr> <td>Embauba</td> <td>0.75</td> <td>1.00</td> <td>0.86</td> <td>6</td> </tr> <tr> <td>Jeriva</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>4</td> </tr> <tr> <td>Mulungu</td> <td>0.91</td> <td>0.91</td> <td>0.91</td> <td>11</td> </tr> <tr> <td>Pitangueira</td> <td>0.80</td> <td>0.67</td> <td>0.73</td> <td>6</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td>0.86</td> <td>43</td> </tr> <tr> <td>macro avg</td> <td>0.87</td> <td>0.87</td> <td>0.86</td> <td>43</td> </tr> <tr> <td>weighted avg</td> <td>0.87</td> <td>0.86</td> <td>0.86</td> <td>43</td> </tr> </tbody> </table>		precision	recall	f1-score	support	Aroeira	0.78	0.88	0.82	8	Capororoca	1.00	0.75	0.86	8	Embauba	0.75	1.00	0.86	6	Jeriva	1.00	1.00	1.00	4	Mulungu	0.91	0.91	0.91	11	Pitangueira	0.80	0.67	0.73	6	accuracy			0.86	43	macro avg	0.87	0.87	0.86	43	weighted avg	0.87	0.86	0.86	43
	precision	recall	f1-score	support																																															
Aroeira	0.78	0.88	0.82	8																																															
Capororoca	1.00	0.75	0.86	8																																															
Embauba	0.75	1.00	0.86	6																																															
Jeriva	1.00	1.00	1.00	4																																															
Mulungu	0.91	0.91	0.91	11																																															
Pitangueira	0.80	0.67	0.73	6																																															
accuracy			0.86	43																																															
macro avg	0.87	0.87	0.86	43																																															
weighted avg	0.87	0.86	0.86	43																																															
Principais perdas	<p style="text-align: center;">Prediction/Actual/Loss/Probability</p> <div style="display: flex; flex-wrap: wrap; justify-content: space-around;"> <div style="text-align: center; margin: 5px;"> <p>Pitangueira/Mulungu / 4.23 / 0.98</p>  </div> <div style="text-align: center; margin: 5px;"> <p>Aroeira/Pitangueira / 2.86 / 0.82</p>  </div> <div style="text-align: center; margin: 5px;"> <p>Embauba/Capororoca / 2.53 / 0.56</p>  </div> <div style="text-align: center; margin: 5px;"> <p>Mulungu/Pitangueira / 2.36 / 0.84</p>  </div> </div>																																																		
Predição	<i>Upload</i>	Imagem de embauba.																																																	

		Predição	 <p>nguar ue, realizando prediçã...</p> <p>Isto é um(a) Embauba.</p>
--	--	----------	--

Tabela 19 - Resultados do treinamento dos modelos.

O modelo 1 foi o mais simples, rodado com a arquitetura resnet18 com 10 epochs concluindo a execução em apenas 8 minutos. No *transfer learning* já obteve uma acurácia de 95%. Ao analisar a matriz de confusão observa-se que apenas 2 itens foram confundidos. Vendo a acurácia por categoria, apenas uma classe obteve resultados abaixo de 90%. Nesse caso não se executou o *fine tuning*.

Com a arquitetura vg_16bn, utilizando o processo de aumentações e treinando o modelo por 5 epochs, o modelo 2 concluiu a execução em 12 minutos. No transfer learning obteve uma acurácia de 81%. Na matriz de confusão fica possível ver que apenas 14 itens foram confundidos. Neste caso, foi executado o fine tuning, demorando mais 8 minutos resultando em uma acurácia de 84% e resultando em apenas 8 itens errados na matriz de confusão.

O modelo mais elaborado foi o Modelo 3, que usou a arquitetura densenet121 e foi treinado por 20 epochs. Foram obtidos os melhores resultados, como mais de 90% de acurácia e apenas 6 itens na matriz de confusão.

Em geral a ferramenta se saiu bem e consegui cumprir tudo o que foi proposto, realizando a tarefa de classificação de imagens single label. A ferramenta contemplou modelos simples e mais complexos, com diferentes parâmetros e passos opcionais, conseguindo consequentemente demorar menos ou mais tempo dependendo do que foi selecionado.

5. Avaliação do Vulcan

5.1. Definição da avaliação

Para avaliar a qualidade da ferramenta, foi realizada uma avaliação preliminar na forma de um painel de especialistas. O objetivo foi analisar a qualidade da camada visual em termos de utilidade, adequação funcional, eficiência de desempenho e usabilidade do ponto de vista de professores e alunos no contexto da educação em computação. De acordo com o foco no nível médio, foram incluídos professores do ensino médio. No entanto, devido ao fato de que atualmente a maioria dos alunos do ensino médio não possui nenhum conhecimento sobre ML e, portanto, podem não ser capazes de avaliar tal suporte, foram incluídos neste estudo preliminar alunos de graduação com conhecimento sobre ML. E, tendo em vista que tal interface visual também pode ser utilizada em um momento inicial nos cursos de ML no ensino superior, ampliou o grupo de participantes incluindo também professores do ensino médio.

Com base na ISO / IEC 25010 (2011), ISO / IEC 9241 (2019), TAM (DAVIS, 1989) e SUS (BROOKE, 1996), os fatores de qualidade a serem avaliados são decompostos conforme apresentado na Tabela 20.

Característica	Sub-característica	Avaliação do usuário		
		Questionário		Escala de resposta
		Questionário do professor	Questionário do aluno	
Utilidade		Acho que a ferramenta é útil para ensinar ML no ensino médio	A ferramenta é útil para aprender ML	Escala Likert de 5 pontos (Concordo totalmente, Concordo, Não concordo nem discordo;
		Acho que a ferramenta é útil para ensinar ML para iniciantes na graduação / alunos na graduação		

		Acho que pode ajudar os alunos a aprender conceitos de ML	Acho que pode me ajudar a aprender conceitos conceituais de ML	Discordo; Discordo totalmente)
		Eu acho que pode ser mais útil do que a interface de programação baseada em texto convencional do Jupyter para começar a ensinar o desenvolvimento de um modelo de ML para iniciantes	Eu preferiria começar a aprender o desenvolvimento de um modelo de ML usando essa interface visual do que usando a interface de programação baseada em texto convencional do Jupyter	
Funcionalidade	Compleitude	Existem etapas/partes para ensinar o desenvolvimento da classificação de imagens para iniciantes que não são suportadas pela ferramenta?		Sim, não (se sim, qual?)
	Corretude	Você notou algum erro em relação à funcionalidade da ferramenta?	Você notou algum erro em relação à funcionalidade da ferramenta?	Sim, não (se sim, qual?)
Desempenho	Comportamento de tempo	O desempenho da ferramenta é satisfatório?	O desempenho da ferramenta é satisfatório?	Sim, não (se não, porquê?)
Usabilidade	Eficácia	Você conseguiu desenvolver um modelo de ML para classificação de imagens com a ferramenta?	Você conseguiu desenvolver um modelo de ML para classificação de imagens com a ferramenta?	Sim, não (se não, porquê?)
	Eficiência	Quanto tempo aproximadamente você demorou para desenvolver um modelo de ML usando a ferramenta?	Quanto tempo aproximadamente você demorou para desenvolver um modelo de ML usando a ferramenta?	___ minutos
		Você considera apropriado o tempo que levou para desenvolver o modelo de ML com a ferramenta?	Você considera apropriado o tempo que levou para desenvolver o modelo de ML com a ferramenta?	Sim, Não

		Em comparação com uma versão de programação baseada em texto de um Jupyter notebook, você demorou menos tempo, mais tempo ou o mesmo para desenvolver um ML para classificação de imagens?	Em comparação com uma versão de programação baseada em texto de um Jupyter notebook, você demorou menos tempo, mais tempo ou o mesmo para desenvolver um ML para classificação de imagens?	Menos tempo, igual, mais tempo
	Satisfação	Eu acho que gostaria de usar esse sistema com frequência.	Eu acho que gostaria de usar esse sistema com frequência.	Escala Likert de 5 pontos (Concordo totalmente, Concordo, Não concordo nem discordo; Discordo; Discordo totalmente)
		Eu acho o sistema desnecessariamente complexo.	Eu acho o sistema desnecessariamente complexo.	
		Eu achei o sistema fácil de usar.	Eu achei o sistema fácil de usar.	
		Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	
		Eu acho que as várias funções do sistema estão muito bem integradas.	Eu acho que as várias funções do sistema estão muito bem integradas.	
		Eu acho que o sistema apresenta muita inconsistência.	Eu acho que o sistema apresenta muita inconsistência.	
		Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.	Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.	
		Eu achei o sistema atrapalhado de usar.	Eu achei o sistema atrapalhado de usar.	
		Eu me senti confiante ao usar o sistema.	Eu me senti confiante ao usar o sistema.	
		Eu precisei aprender várias coisas novas antes	Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.	

		de conseguir usar o sistema.		
	Operabilidade	Você acha que a ferramenta de avaliação possui elementos ambíguos ou difíceis de entender?	Você acha que a ferramenta de avaliação possui elementos ambíguos ou difíceis de entender?	Sim, não (se sim, qual?)
Pontos fortes e pontos a melhorar		O que você mais gostou na ferramenta de avaliação?	O que você mais gostou na ferramenta de avaliação?	Aberta
		O que você menos gostou na ferramenta de avaliação?	O que você menos gostou na ferramenta de avaliação?	Aberta
		Mais alguma sugestão?	Mais alguma sugestão?	Aberta

Tabela 20. Visão geral da decomposição das características de qualidade e operacionalização da medição.

O painel de especialistas visa avaliar a qualidade percebida do ponto de vista de professores e alunos. Durante o painel de especialistas, os usuários recebem primeiro uma visão geral básica do objetivo e dos recursos da ferramenta. Em seguida, eles realizam a tarefa de desenvolver um modelo de ML para a classificação de imagens de seis espécies de árvores com um conjunto de dados preparado executando todo o processo de desenvolvimento do modelo de ML.

Os dados são coletados por meio de um questionário pós-teste. Os itens do questionário foram derivados das características de qualidade (Tabela 20). Foram utilizadas diferentes escalas, incluindo uma escala nominal (sim/não), uma escala ordinal, bem como uma escala Likert de 5 pontos dependendo da natureza do item. Além disso, também foi solicitada a identificação de pontos fortes e oportunidades de melhoria aos participantes.

Instruções e formulários de coleta de dados foram disponibilizados online via LimeSurvey.

5.2. Execução da avaliação

A avaliação da ferramenta foi realizada em agosto de 2021. Inicialmente foram convidados 26 alunos de graduação e pós-graduação e professores do ensino médio e ensino superior. Os participantes foram selecionados por amostragem de conveniência junto a membros da iniciativa Computação na Escola/INCoD/INE/UFSC com conhecimento prévio em Aprendizado de Máquina para classificação de imagens. Os participantes foram convidados por e-mail explicando o objetivo da avaliação e garantindo o sigilo. A participação foi voluntária.

Ao final foram obtidas 14 respostas completas, representando uma taxa de resposta de 54%, alguns assumindo mais de um papel como professor do ensino médio e aluno da pós-graduação, por exemplo.

Figura 12 apresenta a distribuição demográfica dos participantes da avaliação. Mais detalhes referentes aos dados coletados são apresentados no apêndice A e também disponível online⁷.

⁷ <https://datastudio.google.com/reporting/78facdda-ff39-436f-972c-9a710f8fc50f>

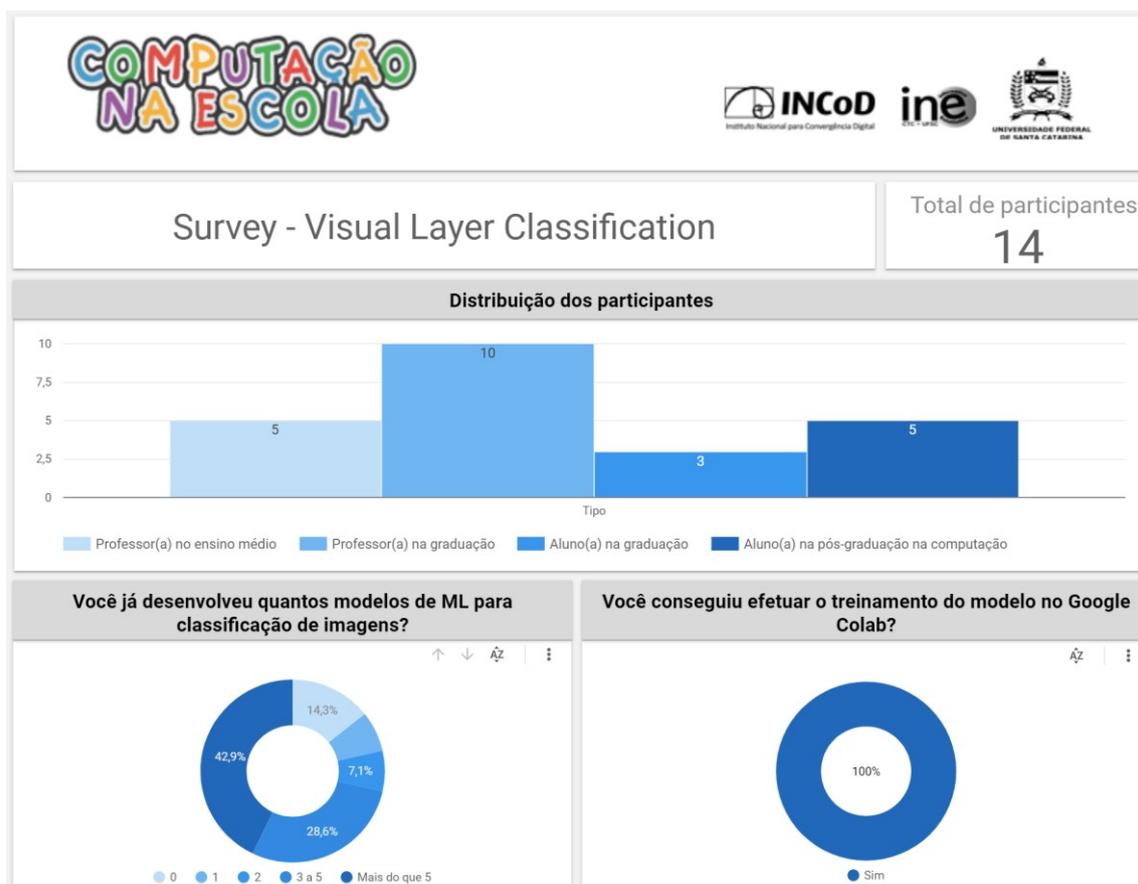


Figura 12 - Visão geral da distribuição demográfica dos participantes.

5.3. Análise dos resultados

Para uma melhor interpretação das informações, as respostas são agrupadas por critério de análise.

5.3.1. Utilidade

Todos os alunos participantes acharam a ferramenta útil para aprender ML, e 9 entre 10 professores acharam a ferramenta útil para ensinar ML no ensino médio. Quando perguntados para ensinar alunos iniciantes na graduação todos os professores acharam a ferramenta útil. A maioria dos participantes acha que pode ajudar a aprender conceitos de ML, além de achar que a ferramenta pode ser mais útil para aprender a desenvolver um modelo de ML em comparação ao ambiente tradicional, com exceção de um aluno (Tabela 21).

Papel	Questão de análise	Discordo totalmente		Discordo		Não concordo nem discordo		Concordo		Concordo totalmente		Total
Professor	Acho que a ferramenta é útil para ensinar ML no ensino médio	0	0%	1	10%	0	0%	5	50%	4	40%	10
Aluno	A ferramenta é útil para aprender ML	0	0%	0	0%	0	0%	5	55,6%	4	44,4%	9
Professor	Acho que a ferramenta é útil para ensinar ML para iniciantes na graduação / alunos na graduação	0	0%	0	0,00%	0	0%	5	50%	5	50%	10
Professor	Acho que pode ajudar os alunos a aprender conceitos de ML	0	0%	0	0,00%	2	20%	3	30%	5	50%	10
Aluno	Acho que pode me ajudar a aprender conceitos conceituais de ML	0	0%	0	0,00%	1	11,1%	6	66,6%	2	22,2%	9
Professor	Eu acho que pode ser mais útil do que a interface de programação baseada em texto convencional do Jupyter para começar a ensinar o desenvolvimento de um modelo de ML para iniciantes.	0	0%	0	0,00%	1	10%	4	40%	5	50%	10
Aluno	Eu preferiria começar a aprender o desenvolvimento de um modelo de ML usando essa interface visual do que usando a interface de programação baseada em texto convencional do Jupyter	0	0%	1	11,11%	2	22,2%	2	22,2%	4	44,4%	9
	Total	0	0%	2	2,9%	6	8,9%	30	44,7%	29	43,2%	67

Tabela 21 - Análise da utilidade da ferramenta.

5.3.2. Funcionalidade

Em geral, a completude e a corretude obtiveram boas avaliações.

Apenas dois participantes apontaram partes não suportadas pela ferramenta, sendo ela a rotulação das imagens. Como definido isso ocorre dentro da ferramenta, mas houve uma falha na comunicação com os participantes e acabando não deixando isso explícito. Mas estes relatos mostram que o processo de rotulação pode ser retrabalhado, e quem sabe esteja dentro da ferramenta em trabalhos futuros. Em relação a erros, poucos usuários relataram problemas, entre eles a probabilidade da predição que muitas vezes acertava a classe mas apresentava um percentual incorreto.

Pergunta	sim		não		total
	count	percentage	count	percentage	
Existem etapas/partes para ensinar o desenvolvimento da classificação de imagens para iniciantes que não são suportadas pela ferramenta?	12	85,71%	2	14,29%	14
Você notou algum erro em relação à funcionalidade da ferramenta?	3	21,43%	11	78,57%	14

Tabela 22 - Análise da funcionalidade da ferramenta.

5.3.3. Desempenho

Todos os participantes acharam o resultado da ferramenta satisfatório, conseguindo executar todas as etapas rapidamente, com exceção das parametrizações que naturalmente aumentam o tempo de execução do treinamento do modelo.

Pergunta	sim		não		total
	cont.	porcentagem	cont.	porcentagem	
O desempenho da ferramenta é satisfatório?	14	100%	0	0%	14

Tabela 23 - Análise do desempenho da ferramenta.

5.3.4. Usabilidade

Em geral, a usabilidade da ferramenta teve um *feedback* muito positivo. Em relação a eficácia 13 entre os 14 participantes conseguiram desenvolver o modelo para classificação de imagens.

Pergunta	sim		não		total
	cont.	porcentagem	cont.	porcentagem	
Você conseguiu desenvolver um modelo de ML para classificação de imagens com a ferramenta?	13	92,86%	1	7,14%	14
Você considera apropriado o tempo que levou para desenvolver o modelo de ML com a ferramenta?	14	100%	0	0%	14

Tabela 24 - Análise da usabilidade da ferramenta.

A ferramenta também se mostrou eficiente, como todos os usuários considerando apropriado o tempo que levou para desenvolver o modelo. Além disso, 11 dos 13 usuários que concluíram o desenvolvimento do modelo terminaram a tarefa em até 30 minutos. Também 10 dos 14 participantes acham que o tempo levado para realizar todo o processo é mais rápido do que em um ambiente baseado em texto.

Pergunta	Até 30 min		Até 60 min		total
	cont.	porcentagem	cont.	porcentagem	
Quanto tempo aproximadamente você demorou para desenvolver um modelo de ML usando a ferramenta?	11	84,62%	2	15,38%	13

Tabela 25 - Análise do tempo de treinamento da ferramenta.

Pergunta	menos tempo		mesmo ou mais tempo		total
	cont.	porcentagem	cont.	porcentagem	
Em comparação com uma versão de programação baseada em texto de um Jupyter notebook, você demorou menos tempo, mais tempo ou o mesmo para desenvolver um ML para classificação de imagens?	10	71,43%	4	28,57%	14

Tabela 26 - Análise da comparação da ferramenta com ambiente tradicional.

Para medir a satisfação dos usuários foi aplicado o questionário SUS (Brooke, 1996), e como resultado foi obtida a boa média de 80,3%. Individualmente a nota mais

baixa foi de 72,5%, ainda indicando uma boa satisfação. A questão individual com a média mais baixa foi a que foi perguntado se é possível aprender como usar esse sistema rapidamente. A maior pontuação de uma questão individual foi referente a consistência do sistema, com nenhum participante indicando inconsistências.

Pergunta	Discordo totalmente	Discordo	Não concordo nem discordo	Concordo	Concordo totalmente	Pontuação SUS
Eu acho que gostaria de usar esse sistema com frequência.	0 (0%)	0 (0%)	3 (21,4%)	7 (50%)	4 (28,6%)	7,68
Eu acho o sistema desnecessariamente complexo.	5 (35,7%)	8 (57,1%)	1 (7,1%)	0 (0%)	0 (0%)	8,21
Eu achei o sistema fácil de usar.	0 (0%)	0 (0%)	0 (0%)	10 (71,4%)	4 (28,6%)	8,21
Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	5 (35,7%)	6 (42,9%)	2 (14,3%)	1 (7,1%)	0 (0%)	7,68
Eu acho que as várias funções do sistema estão muito bem integradas.	0 (0%)	0 (0%)	2 (14,3%)	7 (50%)	5 (35,7%)	8,04
Eu acho que o sistema apresenta muita inconsistência.	9 (64,3%)	5 (35,7%)	0 (0%)	0 (0%)	0 (0%)	9,11
Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.	0 (0%)	2 (14,3%)	2 (14,3%)	6 (42,9%)	4 (28,6%)	7,14
Eu achei o sistema atrapalhado de usar.	4 (28,6%)	9 (64,3%)	1 (7,1%)	0 (0%)	0 (0%)	8,04
Eu me senti confiante ao usar o sistema.	0 (0%)	0 (0%)	3 (21,4%)	6 (42,9%)	5 (35,7%)	7,86
Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.	6 (42,9%)	7 (50%)	1 (7,1%)	0 (0%)	0 (0%)	8,39
Total SUS						80,36

Tabela 27 - Análise do questionário SUS.

Em relação a operabilidade ficou evidente que apesar de os participantes conseguirem executar o processo em geral faltavam informações sobre como fazer, o

que exatamente estavam fazendo ou como interpretar as entradas de dados e retornos.

pergunta	sim		não	
	Você acha que a ferramenta de avaliação possui elementos ambíguos ou difíceis de entender?	5	35,71%	9

Tabela 28 - Análise da operabilidade da ferramenta.

5.4. Discussão

Pontos fortes.

O *feedback* dos participantes fornece uma primeira indicação que o *Visual Layer Classification* pode ser útil, funcional, eficiente em desempenho e ter boa usabilidade. Como pontos fortes, os avaliadores destacaram, principalmente, a simplicidade e facilidade de desenvolver um modelo de ML de ponta a ponta sem necessariamente ter experiência com códigos e também sua completude. Os participantes elogiaram a praticidade do fluxo do processamento, podendo contemplar etapas que são mais complicadas ou adicionais caso desejem ou não, como realizar as aumentações com transformações, definir como quais métricas serão utilizadas e acompanhadas bem além das várias opções de visualização dos resultados. A possibilidade de poder realizar muitos testes variando as opções disponíveis também foi elogiada, uma vez que se pode demonstrar bem o ciclo de *use-modify-create*.

Sugestões de melhoria.

As principais sugestões de melhoria foram principalmente relacionadas à falta de informações extras e explicações conceituais em geral sobre o fluxo, entradas e saídas. Parte disso já foi resolvido com a implementação da documentação externa contendo mais detalhes sobre cada etapa do processo em geral. Cada seção agora tem um ponto de interrogação com um link direto a seção da documentação referente

aquela parte do processo, cobrindo todos os elementos da tela. Outro detalhe mencionado foi a respeito dos elementos visuais que poderiam ser melhorados, ficando mais polidos ou apresentando barras de progresso para facilitar o andamento de determinados processos. Além disso, foi sugerida a possibilidade de mais interação dos alunos com o código.

Pergunta	Comentários dos professores	Comentários dos alunos
<p>O que você mais gostou na ferramenta</p>	<ul style="list-style-type: none"> ● passo a passo já estabelecido, DS pronto, integração com métricas. ● Simplicidade. ● Praticidade no fluxo do processamento. ● Passo a passo do processo de treinamento de forma parametrizada. ● O que eu mais gostei foi da facilidade de desenvolvimento de um modelo de ML utilizando interface gráfica. Os passos estão bem definidos e a apresentação dos resultados é clara e de fácil entendimento. ● Fácil entendimento ● Expor vários ajustes ● Ela possibilita usar várias técnicas para transformação e aumento do dataset. ● A ferramenta é bem completa e fácil de usar. Gostei também do formato de interface que permite atingir aqueles que não tem um conhecimento de programação. Parabéns pelo trabalho! ● A disposição e interação visual dentro do Jupyter Notebook facilitando o processo de classificação de imagens. Permitiu uma fácil e rápida visão do processo como um todo. 	<ul style="list-style-type: none"> ● Ter todo o sistema em ambiente gráfico, sem precisar usar nenhum código. ● Facilidade de recriar/retestar a rede com diferentes configurações ● Direta ao ponto, simples, concisa. Mensagens de sucesso ajudam bastante a entender que a etapa foi concluída. ● A facilidade de somente clicar em um botão/inserir um texto no widget e já visualizar o resultado de uma forma concisa e compacta.

<p>O que você menos gostou na ferramenta</p>	<ul style="list-style-type: none"> ● adaptação inicial, poucas informações sobre os conceitos ● Scroll interno da ferramenta pode gerar confusão para usuários com menos experiência. Senti falta de um botão para limpar os elementos de predição no upload. ● O design visual ainda necessita ser melhorado, em especial os botões e a integração ao Google Drive está complicada. ● Não é exatamente que "não gostei", mas acredito que poderia ter algumas explicações mais conceituais ● Não ter acesso a imagens para poder visualizar os resultados de uma predição sem fazer upload ● Não que não tenha gostado, mas levei uns 2 minutos para entender (fiquei lendo o código dentro da primeira célula) que a proposta era utilizar o JupyterNotebooks totalmente de forma visual, apenas pelas abas e botões. O que agora me parece excelente! ● Gostei de tudo :D ● Embora o código esteja disponível, ele não é obrigatório (entendimento ou mesmo manipulação/modificação) para o uso da ferramenta. Entretanto, abstrair os procedimentos pode, em um determinado nível - e no meu entendimento - distanciar o usuário dos conceitos básicos de ML. Acredito que uma maior interação com a ferramenta possa ajudar no processo. ● Acho que falta um feedback visual para o resultado da aplicação das técnicas de transformação/aumentação. ● - Seria interessante uma barra de progresso ou mensagem de "aguarde ..." no passo "Ver 	<ul style="list-style-type: none"> ● O conteúdo pode ser aprimorado para tornar a ferramenta e as etapas mais claras. ● O ambiente gráfico não coube na célula. Em alguns momentos tive que rolar a tela para cima e para baixo para achar o que precisava. ● Nada a comentar. ● Nada.
--	--	--

	<p>batch”.</p> <ul style="list-style-type: none"> - Facilitaria o uso se somente fosse habilitada a execução do próximo passo quando o passo anterior tivesse sido executado com sucesso. - A célula de “Transformações e Aumentações” ficou com um tamanho muito grande. - “unfreeze” poderia ser traduzido para “descongelar”, ou soaria estranho? - No passo de “Data Cleaning”, foi gerada uma barra horizontal gigante para a direita. Esse formato dificulta a visualização e a navegação. Também as imagens ficaram coladas umas às outras. 	
--	--	--

Tabela 29 - Comentários dos alunos e professores.

Ameaças a validade da avaliação.

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. Embora os participantes foram 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 conhecimento prévio dos participantes. Assim, indicando a necessidade de estudos futuros com maior número de participantes do perfil específicos do público alvo. Também é importante ressaltar que a ferramenta ainda estava em desenvolvimento quando foi submetida ao painel de especialistas, logo podendo apresentar erros e inconsistências durante os testes.

Além disso, o tamanho da amostra pode comprometer a generalização dos resultados. O estudo foi baseado em um total de 14 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).

6. Conclusão

O objetivo geral deste trabalho foi desenvolver uma ferramenta visual em um notebook para o ensino de Machine Learning no ensino médio, abordando desde a análise de requisitos até a exportação do modelo para implantação. Para isso foi realizada uma síntese da fundamentação teórica (O1). Também foi analisado o estado da arte em relação a ferramentas para visualizar o processo de desenvolvimento de sistemas de Machine Learning para o ensino médio, mostrando que existem poucas ferramentas com essa finalidade (O2). Assim, alinhado com a fundamentação teórica e a análise do estado da arte foi elaborado um modelo conceitual da ferramenta de um classificador single label completo no colab, dando início ao desenvolvimento da biblioteca visual para suportar o processo de desenvolvimento de sistemas de Machine Learning para o ensino médio (O3). Os resultados da avaliação da ferramenta fornecem uma indicação inicial da sua utilidade, funcionalidade, desempenho e usabilidade bem positiva (O4).

Assim, é esperado uma contribuição importante para o ensino de ML nas escolas brasileiras, buscando aproximar e viabilizar o contato e aprendizagem dos alunos por meio de uma ferramenta visual para que mesmo sem conhecimento prévio de programação consiga começar a aprender IA/ML e assim contribuir no seu progresso na aprendizagem como um todo.

Como trabalhos futuros, recomenda-se a implementação de mais customizações em relação ao processo de ML em geral, uma vez que existem infinitas possibilidades e/ou necessidades mais específicas, mas entre elas principalmente a entrada dos arquivos por outra fonte que não o Google Drive e também outro método de rotulação das imagens, ambos internos da ferramenta. Também seria interessante retrabalhar a interface visual mais focada na interação entre os objetos da biblioteca

ipywidgets e o Colab, abrindo mais possibilidades para boas práticas de interfaces visuais ou mais interatividade com os elementos durante o fluxo. Outra oportunidade de melhoria seria a construção de uma unidade instrucional completa para complementar a ferramenta, extraíndo todo o potencial dela e focando em como ensinar os conceitos da melhor forma aos alunos.

REFERÊNCIAS

- ADAMS, R. D. **Machine learning algorithm confirms 50 new exoplanets in historic first.** 2020. Disponível em: <https://www.techrepublic.com/article/machine-learning-algorithm-confirms-50-new-exoplanets-in-historic-first/>. Disponível em:
- AGASSI, A., EREL, H., YEHOSHUA WALD, I., ZUCKERMAN, O. **Scratch Nodes ML: A Playful System for Children to Create Gesture Recognition Classifiers.** 2019. In Proceedings of the Conference on Human Factors in Computing Systems, ACM, 1–6.
- AI4ALL. **Big Idea #3: Learning.** 2020. Disponível em: https://drive.google.com/file/d/1QL6I_I5cdNTVnYBIZ3_Lxur2DgFjmG_d/view. Acesso em: 15 nov. 2020.
- 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-accessibleai-education-online-with-the-support-of-google-org-3a6360c135c9>. Acesso em: 07 set. 2020.
- AMAZON. **Amazon Machine Learning, AWS Documentation.** 2019. Disponível em: <https://docs.aws.amazon.com/machine-learning/latest/dg/building-machine-learning.html>. Acesso em: 13 out. 2020.
- Barba, L. A. (2019) **Teaching and Learning with Jupyter.** Disponível em: <https://jupyter4edu.github.io/jupyter-edu-book/>
- BARREIRA, C. M. et al. **Automated large artery occlusion detection in stroke imaging-paladin study.** 2018. Stroke 49, AWP61
- BART, A. C., TIBAU, J., TILEVICH, E., SHAFFER, C. A., KAFURA, D. **BlockPy: An Open Access Data-Science Environment for Introductory Programmers.** 2017. Computer, 50(5), 18-26.
- Basili, V., G. Caldiera and H. D. Rombach. "The Goal Question Metric Approach." (1994).
- BAU, D., GRAY, J., KELLEHER, C., SHELDON, J., TURBAK, F. **Learnable programming: blocks and beyond.** 2017. Communications of the ACM, 60(6), 72-80.
- BEGEL, A. **LogoBlocks: A Graphical Programming Language for Interacting with the World.** 1996. Cambridge, MA: Electrical Engineering and Computer Science Department, MIT.
- BEMLEY, J. L. **Neural networks for precollege students.** 1999. In IJCNN'99. International Joint Conference on Neural Networks, 6, 4422-4427
- BONAR, J., LIFFICK, B. W. **A visual programming language for novices.** 1987. In S. K. Chang, ed. Principles of Visual Programming Systems. Prentice-Hall
- Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.), Usability Evaluation in Industry. London: Taylor and Francis.
- BURNETT, M. M., BAKER, M. J. **A Classification System for Visual Programming Languages.** 1994. Journal of Visual Languages and Computing, 5, 287-300.
- Çakiroğlu, Ü., Suiçmez, S. S, Kurtoğlu, Y. B., Sari, A., Yıldız, S., Öztürk, M. (2018). Exploring perceived cognitive load in learning programming via Scratch. Research in Learning Technology, 26.

CARNEY, M. ET AL. **Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification**. 2020. In Proceedings of Conference on Human Factors in Computing Systems, ACM.

COLAB, Google. **Dados externos: arquivos locais, Drive, Planilhas e Cloud Storage**. 2021. Disponível em: <https://colab.research.google.com/notebooks/io.ipynb#scrollTo=c2W5A2px3doP>. Acesso em: 29 ago. 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, [S.L.], v. 34, n. 6, p. 428-431, dez. 2007. FapUNIFESP (SciELO).

CSTA. **CSTA K-12 Computer Science Standards**. 2017. Disponível em: <http://www.csteachers.org/standards>. Acesso em: 20 set. 2020.

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

DEMŠAR, J. ET AL. **Orange: data mining toolbox in Python**. 2013. *Journal of Machine Learning Research*, 14, 2349-235

DRUGA, S. **Growing up with AI: Cognimates: from coding to teaching machines**. 2018. Master thesis, MIT, USA

DRUGA, S., VU, S. T., LIKHITH, E., QIU, T. **Inclusive AI literacy for kids around the world**. 2019. In Proceedings of FabLearn. ACM, 104–111. DOI:10.1145/3311890.3311904

DUDLEY, J. J., KRISTENSSON, P. O. **A Review of User Interface Design for Interactive Machine Learning**. 2018. *ACM Transactions on Interactive Intelligent Systems*, 8(2), Article 8.

ESTEVA, A., ROBICQUET, A., RAMSUNDAR, B., KULESHOV, V., DEPRISTO, M., CHOU, K., CUI, C., CORRADO, G., THRUN, S., DEAN, J. **A guide to deep learning in healthcare**. 2019. *Nature Medicine*, [S.L.], v. 25, n. 1, p. 24-29. Springer Science and Business Media LLC. Disponível em: <https://www.nature.com/articles/s41591-018-0316-z>. Acesso em: 03 nov. 2020.

FAST.AI. **fast.ai releases new deep learning course, four libraries, and 600-page book**. 2020. Disponível em: <https://www.fast.ai/2020/08/21/fastai2-launch/>. Acesso em: 13 out. 2020.

FAST.AI. **Fast.ai**. 2021. Disponível em: <https://www.fast.ai/>. Acesso em: 29 ago. 2021.

FRASER, N. **Ten things we've learned from Blockly**. 2015. In Proceedings of the 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond). 49–50.

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410–8415.

GODEC, P. ET AL. **Democratized image analytics by visual programming through integration of deep models and small-scale machine learning**. 2019. *Nature Communications*, 10, Article 4551. DOI:10.1038/s41467-019-12397-x

GOODFELLOW, I., BENGIO, Y., COURVILLE, A. **Deep Learning**. 2016. Cambridge: MIT Press.

GOOGLE, Teachable Machine. **About**. 2020. Disponível em: <https://teachablemachine.withgoogle.com/>. Acesso em: 20 set. 2020.

GOOGLE. **Google Colaboratory**. Colab. 2021. Disponível em: <https://colab.research.google.com/>. Acesso em: 18 ago. 2021.

GRESSE VON WANGENHEIM, C., HAUCK, J. C. R., PACHECO, F. S., BUENO, M. F. B. **Visual Tools for Teaching Machine Learning in K-12: A Systematic Mapping**. 2020

GRESSE VON WANGENHEIM, C.; HAUCK, J. C. R.; PACHECO, F. S.; BERTONCELI BUENO, M. F. **Visual Tools for Teaching Machine Learning in K-12: A Ten-Year Systematic Mapping, Education and Information Technologies**, 2021.

Disponível em: <https://doi.org/10.1007/s10639-021-10570-8> Acesso em: 26 ago. 2021.

Hakim, C. (1987). *Research Design: Strategies and Choices in the Design of Social Research*, Contemporary Social Research Series 13. London: Allen and Unwin.

HARVEY, B., MÖNIG, J. **Bringing “no ceiling” to Scratch: Can one language serve kids and computer scientists?**. 2010. In J. Clayson & I. Kalas, eds. *Proceedings of Constructionism 2010 Conference*. 1–10.

HAUCK, M., MACHHAMER, R., CZENKUSCH, L., GOLLMER, K., DARTMANN, G. **Node and Block-Based Development Tools for Distributed Systems with AI Applications**. 2019. *IEEE Access*, 7, 143109-143119, 2019. doi: 10.1109/ACCESS.2019.2940113

HAUTEA, S., DASGUPTA, S., HILL, B. M. **Youth perspectives on critical data literacies**. 2017. In *Proceedings of the Conference on Human Factors in Computing Systems*, ACM, 919–930.

HILS, D. D., **Visual languages and computing survey: Data flow visual programming languages**. 1992. *J. Vis. Lang. Comput.*, vol. 3, no. 1, pp. 69–101, Mar.

HITRON, T., Y. ORLEV, I. WALD, A. SHAMIR, H. EREL, AND O. ZUCKERMAN. **Can Children Understand Machine Learning Concepts? The Effect of Uncovering Black Boxes**. 2019. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Paper 415, 1–11.

HU, Q., MA, L., ZHAO, J. **DeepGraph: A PyCharm Tool for Visualizing and Understanding Deep Learning Models**. 2018. In *Proceedings of the 25th Asia-Pacific Software Engineering Conference*, Nara, Japan, 628-632

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

Idrees, M., Aslam, F., Shahzad, K., Sarwar, S. M.. (2018). *Towards a Universal Framework for Visual Programming Languages*. *Pakistan Journal of Engineering and Applied Sciences*.

IPYWIDGETS. **Ipywidgets**. 2021. Disponível em: <https://ipywidgets.readthedocs.io/en/latest/>. Acesso em: 29 ago. 2021.

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: 07 set. 2020.

JING, M. **China looks to school kids to win the global AI race**. 2018. Disponível em: <https://www.scmp.com/tech/china-tech/article/2144396/china-looks-school-kids-winglobal-ai-race> Acesso em: 25 set. 2020.

JOHNSTON, W. M., HANNA, J. R. P., MILLAR, R. J. **Advances in Dataflow Programming Languages**. 2004. *ACM Comput. Surv.*, vol. 36, no. 1, pp. 1–34, Mar.

JUPYTER. **Jupyter Notebook**. 2021. Disponível em: <https://jupyter.org/about>.

- KAHN, K. M., WINTERS, N. **Child-Friendly Programming Interfaces to AI Cloud Services**. 2017. In: Lavoué É. et al. (eds) *Data Driven Approaches in Digital Education*. Lecture Notes in Computer Science, vol 10474. Springer, Cham.
- KAHN, K. **Three Interactions between AI and Education**. 1977. *Machine Intelligence*, 8.
- KAHN, K., LU, Y., ZHANG, J., WINTERS, N., GAO, M. **Deep learning programming by all**. Proc. of the Conference on Constructionism. 2020. Dublin, Ireland
- KANDLHOFER, M., STEINBAUER, G., HIRSCHMUGL-GAISCH, S., HUBER, P. **Artificial Intelligence and Computer Science in Education: From Kindergarten to University**. 2016. Proc. of IEEE Frontiers in Education Conference, Erie, PA, USA
- KIM, B., GLASSMAN, E., JOHNSON, B., SHAH, J. **iBCM: Interactive Bayesian Case Model Empowering Humans via Intuitive Interaction**. 2015. Technical Report, MIT Computer Science and Artificial Intelligence Laboratory, USA.
- Kluyver, T. et al. (2016). Jupyter Notebooks-a publishing format for reproducible computational workflows. Proc. of the 20th International Conference on Electronic Publishing., IosPress, 87 – 90.
- Knuth, D. E., Pardo, L. T. (1980). The early development of programming languages. In *A history of computing in the twentieth century*, 197–273.
- KO, A.J., MYERS, B.A. AND AUNG, H.H. **Six learning barriers in end-user programming systems**. 2004. In *Proceedings of the IEEE Symp. Visual Languages and Human Centric Computing*, 199–206.
- KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E. **ImageNet classification with deep convolutional neural networks**. 2012. In F. Pereira, C. J. C. Burges, L. Bottou, & K.Q.Weinberger (Eds.) , *Advances in neural information processing systems*, 25 (pp. 1097–1105). Red Hook, NY: Curran
- LANE, D. **Explaining Artificial Intelligence**. 2018. Hello World, 4.
- LIN, P., VAN BRUMMELEN, J., LUKIN, G., WILLIAMS, R., BRAEZEAL, C. **Zhorai: Designing a Conversational Agent for Children to Explore ML Concepts**. 2020. In *Proceedings of the 10th Symposium on Educational Advances in Artificial Intelligence*, New York, NY USA
- LUCK, S. **Pytorch Lightning Machine Learning Zero To Hero In 75 Lines Of Code**. 2012. Disponível em: <https://towardsdatascience.com/pytorch-lightning-machine-learning-zero-to-hero-in-75-lines-of-code-7892f3ba83c0>. Acesso em: 13 out. 2020.
- Lytle, N., Cateté, V., Boulden, D., Dong, Y., Houchins, J., Milliken, A., Isvik, A., Bounajim, D., Wiebe, E., Barnes, T. (2019). Use, modify, create: Comparing computational thinking lesson progressions for stem classes. *Proc. of the ACM Conference on Innovation and Technology in Computer Science Education*, 395–401.
- M. Perkel, J, M.. Why Jupyter is data scientists' computational notebook of choice. *Nature*, 563, 145-146 (2018)
- M. TEMPEL. **Blocks programming**. 2013. *CSTA Voice* 9, 1 (2013), 3–4
- MALONEY, J. H., RESNICK, M., RUSK, N., SILVERMAN, B., EASTMOND, E. **The Scratch programming language and environment**. 2010. *ACM Trans. Comput. Educ.* TOCE 10, 4, 16.
- MARQUES, L. S. GRESSE VON WANGENHEIM. C. HAUCK, J. C. R. **Introdução à Machine Learning- Desenvolvimento de uma unidade instrucional para ensino de ML na educação básica**. 2020b.

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**, 2020

McCracken, M. et al. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In Working group reports from ITiCSE on Innovation and Technology in Computer Science Education, ACM, 125–180.

MINISTÉRIO DA EDUCAÇÃO. **Censo da Educação 2019**. 2019. Disponível em: <http://portal.inep.gov.br/documents/186968/484154/RESUMO+T%C3%89CNICO+-+CENSO+DA+EDUCA%C3%87%C3%83O+B%C3%81SICA+2019/586c8b06-7d83-4d69-9e1c-9487c9f29052?version=1.0>. Acesso em: 25 nov. 2020.

MIT, App Inventor. **About Us**. 2019. Disponível em: <http://appinventor.mit.edu/explore/about-us.html>. Acesso em: 20 set. 2020.

MIT, Scratch. **Acerca**. 2020. Disponível em: <https://scratch.mit.edu/about/>. Acesso em: 20 set. 2020.

MIT, Snap!. **About**. 2020b. <https://snap.berkeley.edu/about>

MITCHELL, T. M. **Machine Learning**. 1997. McGraw-Hill Education, New York.

MODELZOO. **About**. 2019. Disponível em: <https://modelzoo.co/about>. Acesso em: 13 out. 2020.

NGUYEN, G., DLUGOLINSKY, S., BOBÁK, M., TRAN, V., GARCÍA, Á. L., HEREDIA, I., MALÍK, P., HLUCH, L.. **Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey. Artificial Intelligence Review**. 2019. 52(1), 2019, 77–124. DOI:<https://doi.org/10.1007/s10462-018-09679-z>

ONNX. **ONNX**. 2021. Disponível em: <https://onnx.ai/>. Acesso em: 29 ago. 2021.

PAPERT, S., SOLOMON, C. **Twenty things to do with a computer**. 1971. Artificial. Intelligence Memo Number 248. Massachusetts Inst. of Tech

Pasternak, E., Fenichel, R., Marshall, AN (2017). Tips for Creating a Block Language with Blockly Pro-ceedings of the IEEE Blocks and Beyond Workshop, Raleigh, NC, USA 21–24

PETERSEN, K., VAKKALANKA, S., KUZNIARZ, L. **Guidelines for conducting systematic mapping studies in software engineering: An update**, 2015. Information and Software Technology, Volume 64, Pages 1-18, <https://doi.org/10.1016/j.infsof.2015.03.007>.

Piazentin Ono, Freire, J., Silva, C. T. (2021). Interactive Data Visualization in Jupyter Notebooks. *Computing in Science & Engineering*, 23(2), 99-106, doi: 10.1109/MCSE.2021.3052619.

PYTHON. **Python**. 2021. Disponível em: <https://www.Python.org/>. Acesso em: 29 ago. 2021.

QUEIROZ, R. L., SAMPAIO, F. F., LIMA, C., LIMA, P. M.V. **AI from concrete to abstract: demystifying artificial intelligence to the general public**. 2020.

RAMOS, G., MEEK, C., SIMARD, P., SUH, J., GHORASHI, S. **Interactive machine teaching: a human-centered approach to building machine learned models**. 2020. Human–Computer Interaction

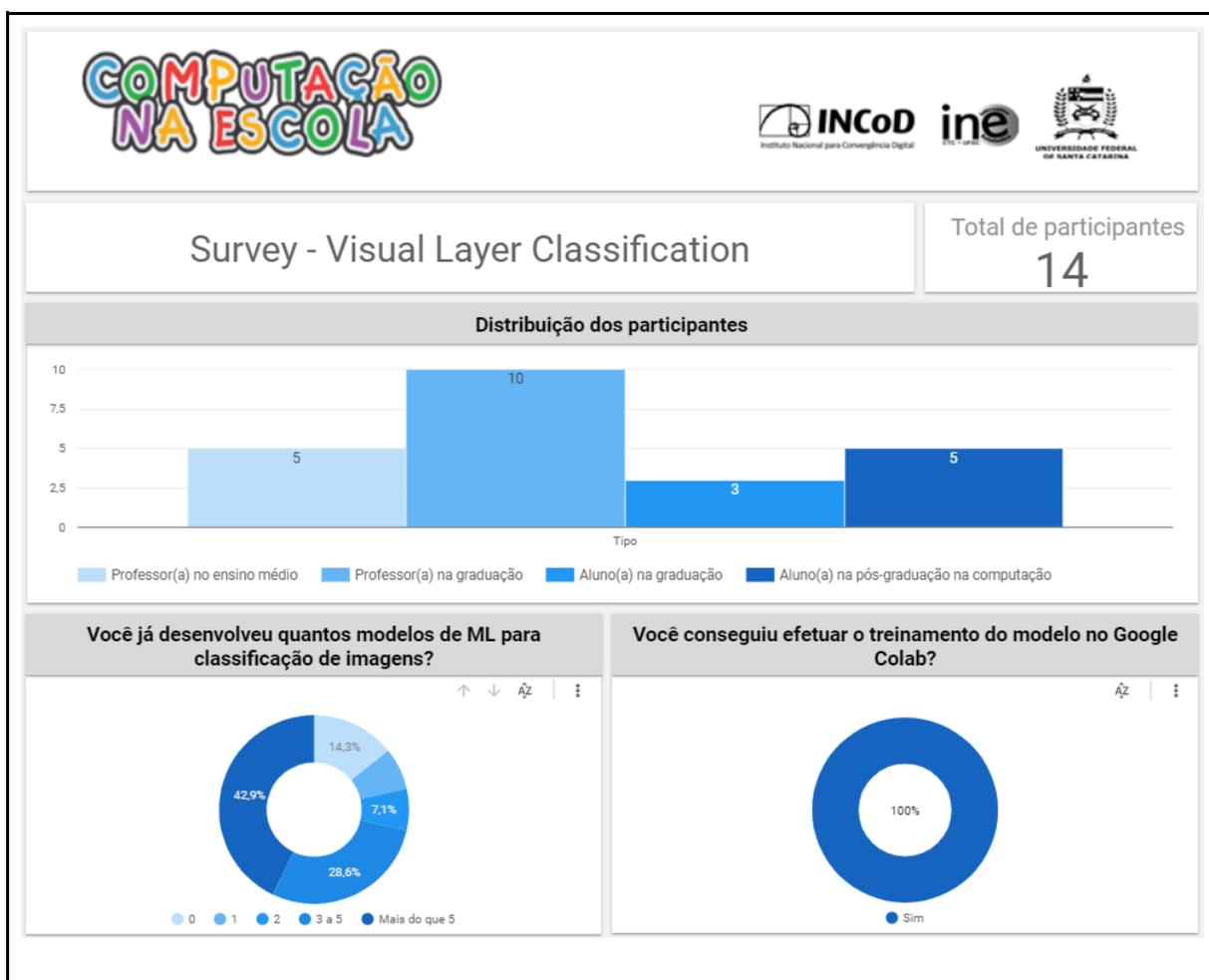
REPENNING, A. **Agentsheets: a tool for building domain-oriented visual programming environments**. 1993. In Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems. ACM, 142–143

- RESNICK, M., BERG, R., EISENBERG, M. **Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation**. 2000. *The Journal of the Learning Sciences*, 9(1), 7–30.
- RIPLEY, B. **Pattern Recognition and Neural Networks**. 2008. University Press, Cambridge.
- ROBY, K. **How AI is helping find a vaccine for COVID-19**. 2020. Disponível em: <https://www.techrepublic.com/article/how-ai-is-helping-find-a-vaccine-for-covid-19/>. Acesso em: 07 set. 2020.
- RODRÍGUEZ GARCÍA, J. D., MORENO-LEÓN, J., ROMÁN-GONZÁLEZ, M., ROBLES, G. **LearningML: A Tool to Foster Computational Thinking Skills Through Practical Artificial Intelligence Projects**. 2020. *Distance Education Journal*, 20(63).
- RODRÍGUEZ-GARCÍA, J. D., MORENO-LEÓN, J., ROMÁN-GONZÁLEZ, M., ROBLES, G. **LearningML: A Tool to Foster Computational Thinking Skills Through Practical Artificial Intelligence Projects**. 2020. *Distance Education Journal*, 20(63).
- ROQUE, R. V. **OpenBlocks: An Extendable Framework for Graphical Block Programming Systems**. 2007. Master's Thesis. Massachusetts Institute of Technology
- ROYAL SOCIETY. **Machine learning: the power and promise of computers that learn by example**. 2017. Disponível em: <https://royalsociety.org/~media/policy/projects/machine-learning/publications/machine-learning-report.pdf>. Acesso em: 07 set. 2020.
- RUSSEL, S., NORVIG, P. **Artificial Intelligence: A Modern Approach, 3ª edição**. New Jersey, NY, USA, 2009
- SAKULKUEAKULSUK, B. ET AL. **Kids making AI: Integrating Machine Learning, Gamification, and Social Context in STEM Education**. 2018. In *Proceedings of the Int. Conference on Teaching, Assessment, and Learning for Engineering*, Wollongong, Australia, 1005-1010.
- Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44(1.2.), 206226.
- Saxena, A., Jain, A., Sener, O., Jami, A., Misra, D. K
- SBC. **Diretrizes para ensino de computação na Educação Básica**. 2020. Disponível em: <https://www.sbc.org.br/educacao/diretrizes-para-ensino-de-computacao-na-educacao-basica>. Acesso em: 20 set. 2020.
- SHEIN, E. **AI is accelerating the move to a touchless world**. 2020. Disponível em: <https://www.techrepublic.com/article/ai-is-accelerating-the-move-to-a-touchless-world/>. Acesso em: 07 set. 2020.
- SLANY, W. **Tinkering with pocket code, a scratch-like programming app for your smartphone**. 2014. In *Proceedings of Constructionism 2014*.
- SMILKOV, D., CARTER, S., SCULLEY, D., VIEGAS, F. B., WATTENBERG, M. **Direct-manipulation visualization of deep networks**. 2017.
- TAMILSELVAM, SRIKANTH, G., PANWAR, N., KHARE, S., ARALIKATTE, R., SANKARAN, A., MANI, S. **A visual programming paradigm for abstract deep learning model development**. 2019. In *Proceedings of the 10th Indian Conference on Human-Computer Interaction (IndiaHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article 16, 1–11.
- TANG, D., UTSUMI, Y., LAO, N. **PIC: A Personal Image Classification Webtool for High School Students**. 2019. In *Proceedings of the IJCAI EduAI Workshop*, Macao, China.
- TECHGIRLZ. **About TechGirlz**. 2020. Disponível em: <https://www.techgirlz.org/about>. Acesso em: 20 set. 2020.
- THARWAT, A. **Classification assessment methods**. 2019. *Applied Computing and Informatics*, in press.

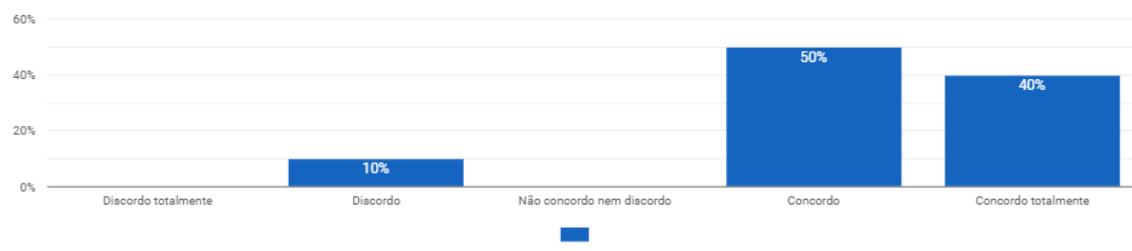
- TISSENBAUM, M., SHELDON, J., ABELSON, H. **From Computational Thinking to Computational Action**. 2019. *Communications of the ACM*, 62(3), 34–36.
- TORREY, L. **Teaching Problem-Solving in Algorithms and AI**. 2012. Proc. of the 3rd Symposium on Educational Advances in Artificial Intelligence, Toronto, Ontario, Canada.
- TOURETZKY, D. S., GARDNER-MCCUNE, C., HUANG, J. **K-12 AI Playground**. 2019c. Proc. of the CSTA Annual Conference, Philadelphia, PA, USA.
- TOURETZKY, D. S., GARDNER-MCCUNE, C., MARTIN, F., SEEHORN, D. **Envisioning AI for K-12: What Should Every Child Know about AI?**. 2019b. Proc. of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, Hawaii, USA.
- TOURETZKY, D. S., GARDNER-MCCUNE, C., MARTIN, F., SEEHORN, D. **K-12 Guidelines for Artificial Intelligence: What Students Should Know**. 2019a. Proc. of the ISTE Conference, Philadelphia, PA, USA
- TRICON INFOTECH. **Top Programming Languages for Machine Learning in 2019**. 2019. Disponível em: <https://medium.com/@triconinfotech/top-programming-languages-for-machine-learning-in-2019-48d4640a4f37> Acesso em: 25 out. 2020.
- UFSC SUSTENTAVEL. **Inventário Florístico do campus Trindade da UFSC**. 2016. Disponível em: <https://ufscsustentavel.ufsc.br/inventario-floristico-do-campus-trindade-da-ufsc/> Acesso em: 25 nov. 2020.
- VAN BRUMMELEN, J., SHEN, J. H., PATTON, E. W. **The Popstar, the Poet, and the Grinch: Relating Artificial Intelligence to the Computational Thinking Framework with Block-based Coding**. 2019. In Proceedings of the Int. Conference on Computational Thinking, Hong Kong, China
- WEINTROP, D. **Block-based Programming in Computer Science Education**. 2019. *Communications of the ACM*, 62(8), 2019, 22–25.
- Weintrop, D., Wilensky, U. (2017). Comparing Block-Based and Text-Based Programming in High-School Computer Science Classrooms. *ACM Transactions on Computing Education*, 18(1), Article 3.
- WILLIAMS, R., PARK, H. W., OH, L., BREAZEAL, C. **PopBots: Designing an Artificial Intelligence Curriculum for Early Childhood Education**. 2019. In Proceedings of the 9th Symposium on Educational Advances in Artificial Intelligence, Menlo Park, CA, USA
- WOHLIN, C. **Guidelines for snowballing in systematic literature studies and a replication in software engineering**. 2014. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. ACM, Article 38, 1–10.
- WOLBER, D., ABELSON, H., SPERTUS, E., LOONEY, L. **App Inventor 2: Create Your Own Android Apps**. 2014. 2nd ed. Beijing: O'Reilly Media.
- WONG, G. K. W., MA, X., DILLENBOURG, P., HUAN, J. **Broadening artificial intelligence education in K-12: where to start?** 2020. *ACM Inroads* 11(1), 20–29.
- WONGSUPHASAWAT, K. ET AL. **Visualizing dataflow graphs of deep learning models in tensorflow**. 2018. *IEEE Transactions on visualization and computer graphics*, 24(1), 1–12, 2018.
- XIE, C., QI, H., MA, L., ZHAO, J. **DeepVisual: A Visual Programming Tool for Deep Learning Systems**. 2019. In Proceedings of the 27th International Conference on Program Comprehension (ICPC), Montreal, QC, Canada, 2019, pp. 130-134.
- ZHU, K. **An Educational Approach to Machine Learning with Mobile Applications**. 2019. M.Eng thesis, MIT, Cambridge, MA, USA
- ZIMMERMANN-NIEFIELD, A., TURNER, M., MURPHY, B., KANE, S. K., SHAPIRO, R. B.

Youth Learning Machine Learning through Building Models of Athletic Moves. In Proceedings of the 18th International Conference on Interaction Design and Children. 2019b. ACM, 121–132.

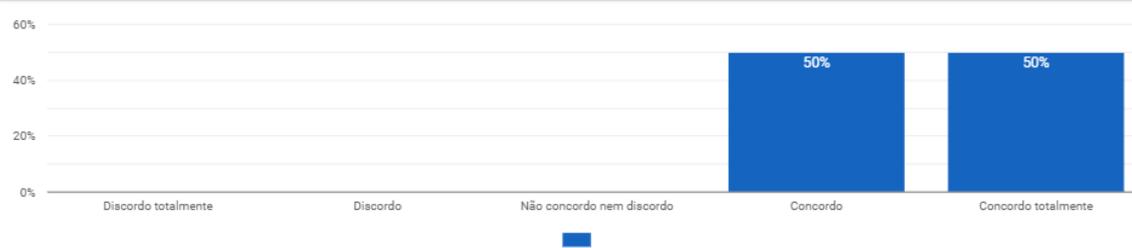
Apêndice A - Resultados do survey



Professor: Acho que a ferramenta é útil para ensinar ML no ensino médio



Professor: Acho que a ferramenta é útil para ensinar ML para iniciantes na graduação / alunos na graduação



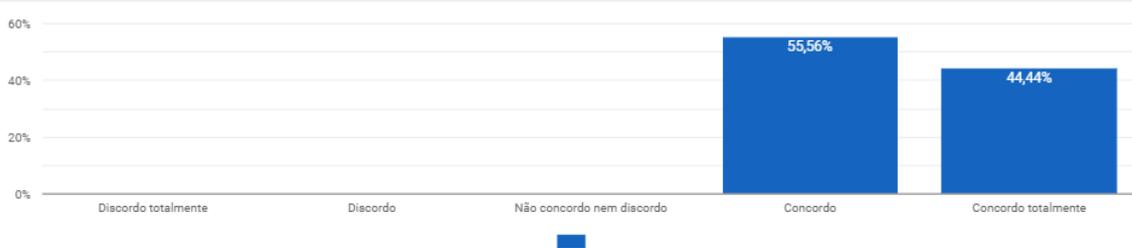
Professor: Acho que pode ajudar os alunos a aprender conceitos de ML

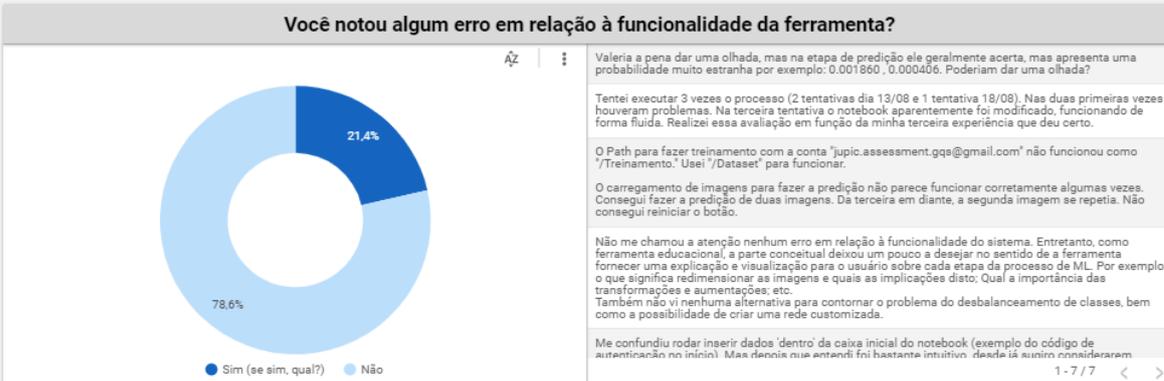
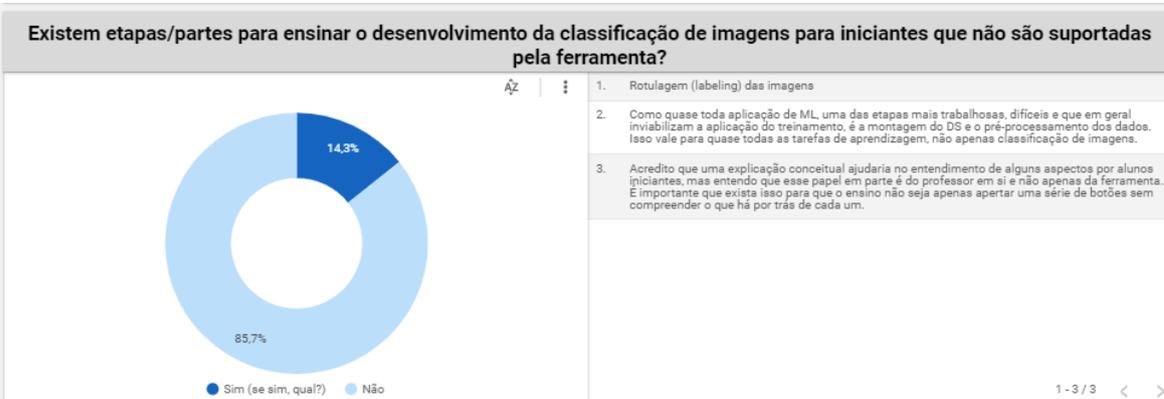
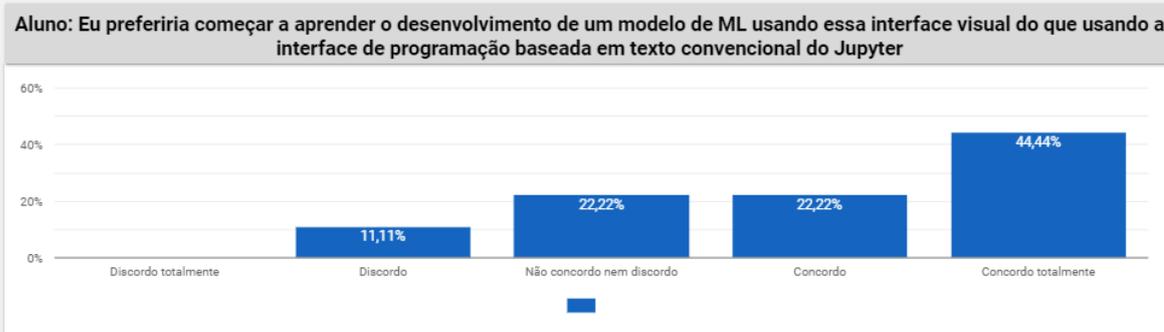
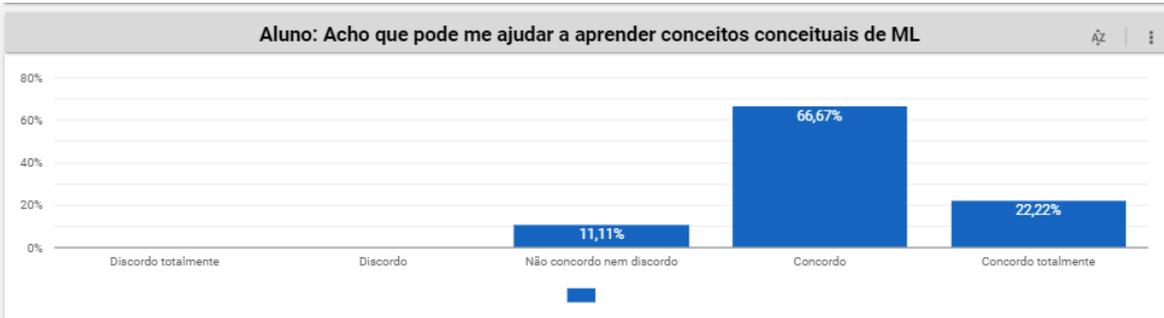


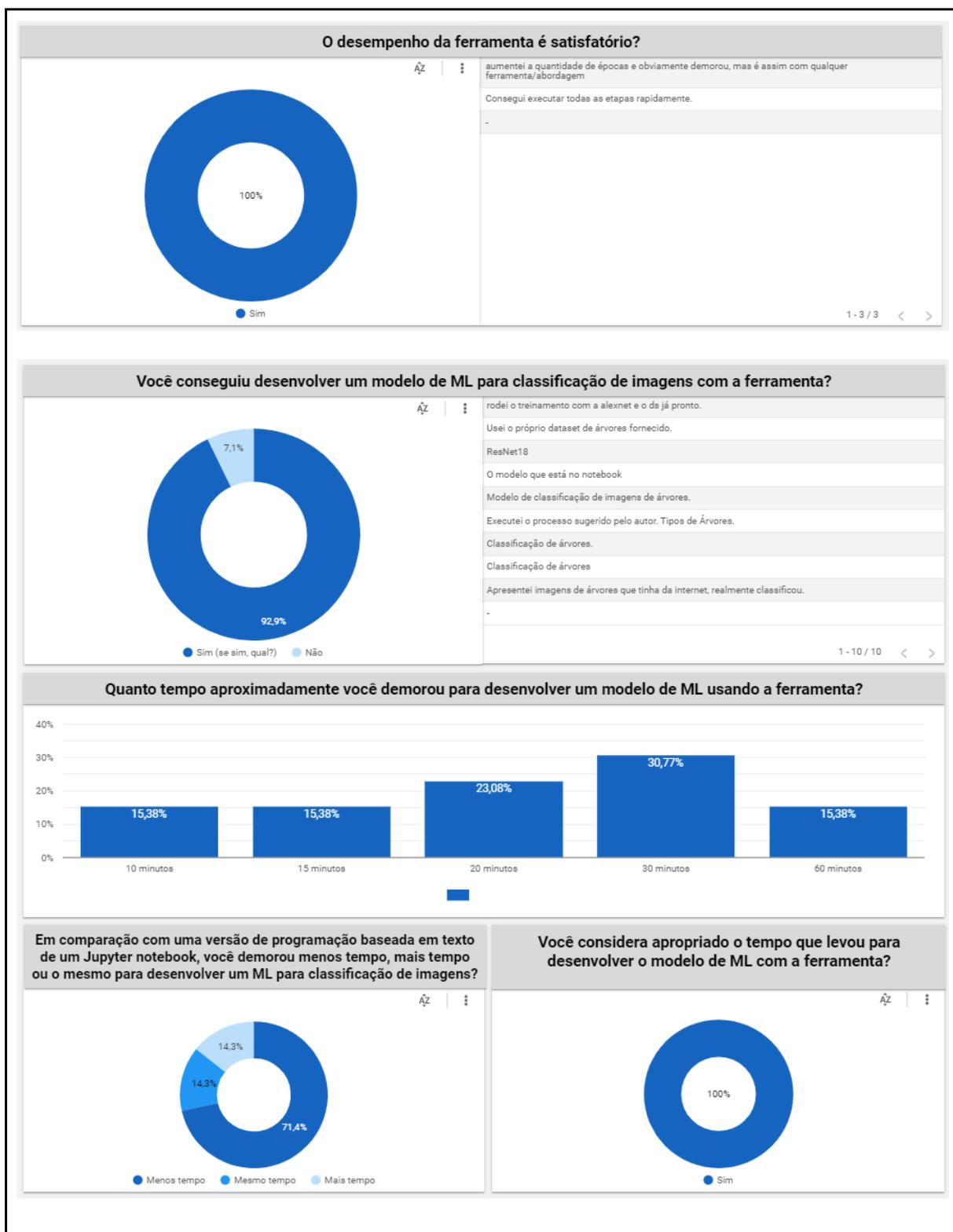
Professor: Eu acho que pode ser mais útil do que a interface de programação baseada em texto convencional do Jupyter para começar a ensinar o desenvolvimento de um modelo de ML para iniciantes

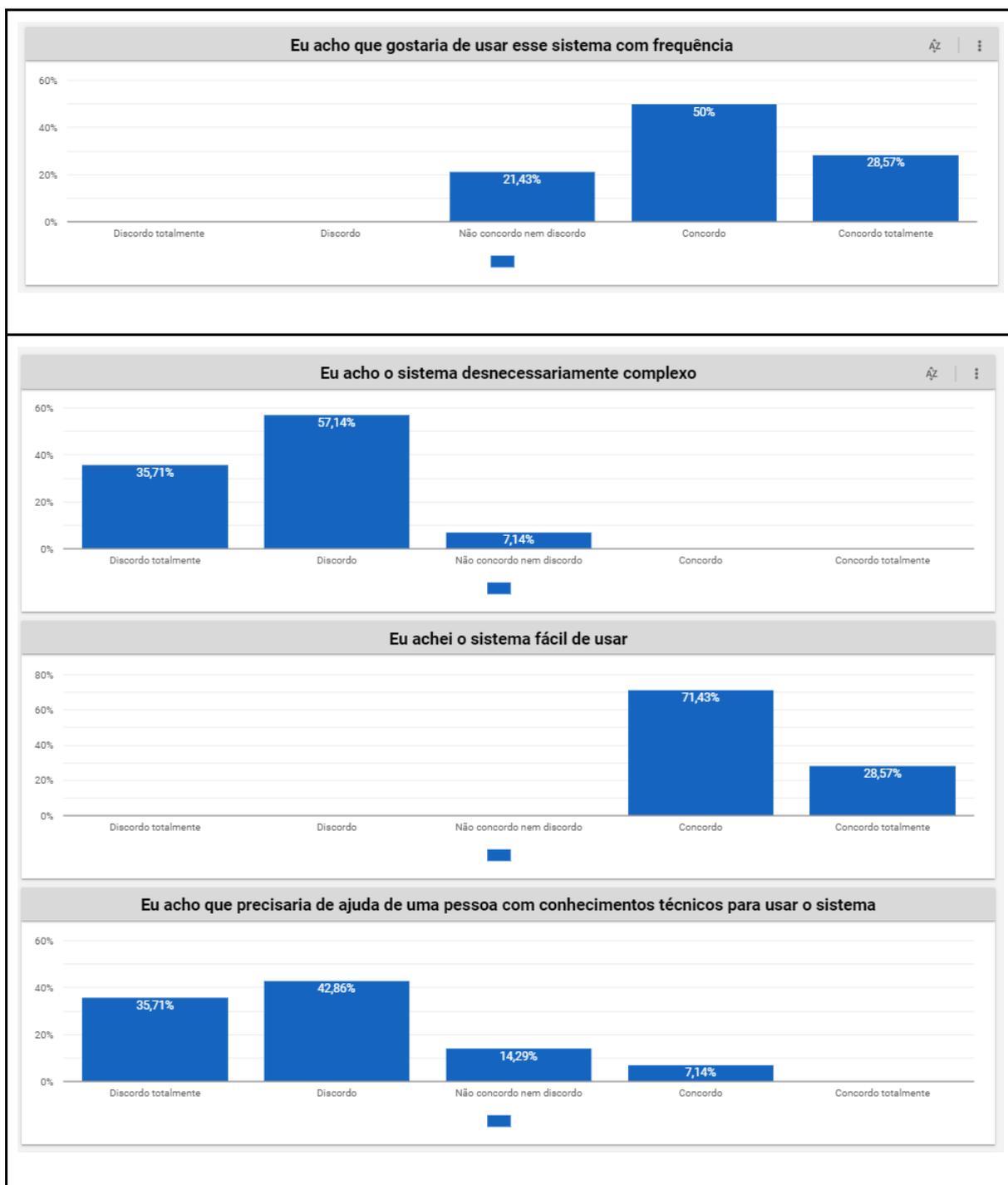


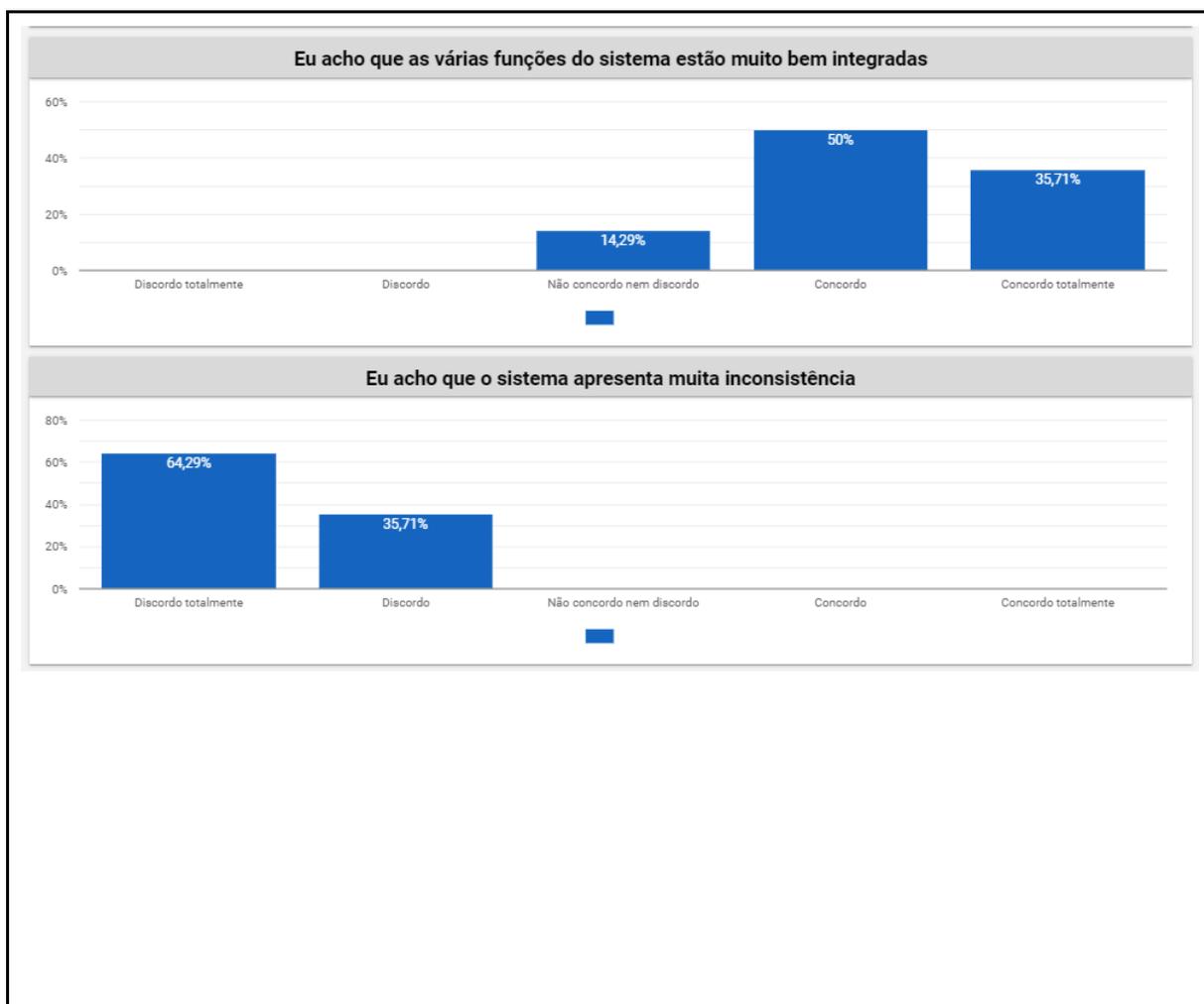
Aluno: A ferramenta é útil para aprender ML

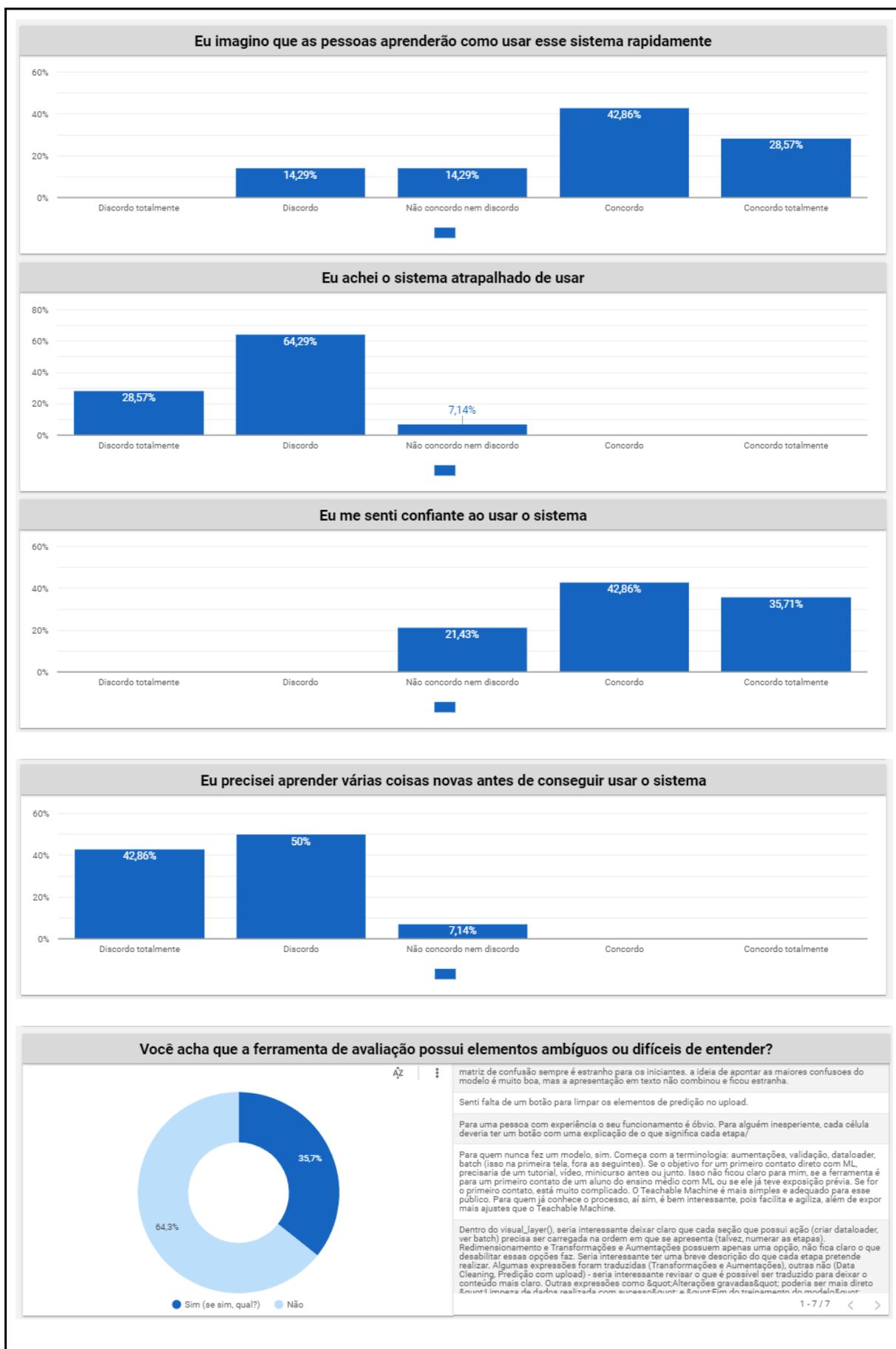












O que você mais gostou na ferramenta de avaliação?	
passo a passo já estabelecido, DS pronto, integração com métricas.	
Ter todo o sistema em ambiente gráfico, sem precisar usar nenhum código.	
Simplicidade.	
Praticidade no fluxo do processamento.	
Passo a passo do processo de treinamento de forma parametrizada.	
O que eu mais gostei foi da facilidade de desenvolvimento de um modelo de ML utilizando interface gráfica. Os passos estão bem definidos e a apresentação dos resultados é clara e de fácil entendimento.	
Fácil entendimento	
Facilidade de recriar/retestar a rede com diferentes configurações	
Expor vários ajustes	
Ela possibilita usar várias técnicas para transformação e aumento do dataset.	
1 - 14 / 14 < >	
O que você menos gostou na ferramenta de avaliação?	Mais alguma sugestão?
adaptação inicial, poucas informações sobre os conceitos	É possível mais interatividade ou resultados intermediários?
Scroll interno da ferramenta pode gerar confusão para usuários com menos experiência. Senti falta de um botão para limpar os elementos de predição no upload.	não.
O design visual ainda necessita ser melhorado, em especial os botões e a integração ao Google Drive está complicada.	já coloquei nos comentários anteriores, parabéns pelo trabalho :-)
O conteúdo pode ser aprimorado para tornar a ferramenta e as etapas mais claras.	Nenhuma, somente verificar a apresentação da predição. Parabens mesmo!!!
O ambiente gráfico não coube na célula. Em alguns momentos tive que rolar a tela para cima e para baixo para achar o que precisava.	Como sugestão, seria interessante uma explicação conceitual sobre cada etapa da criação do modelo, especialmente voltada a alunos iniciantes. Também seria bom um certo cuidado com termos em inglês que são usuais por especialistas na área, mas "grego" para iniciantes. Então seria interessante prover uma explicação para os mesmos, p.s. fora do escopo da ferramenta, mas em relação ao limesurvey, seria bom usar outra ferramenta que não expirasse as respostas (algo bem frustrante), p.ex. Google Forms.
Não é exatamente que "não gostei", mas acredito que poderia ter algumas explicações mais conceituais	Como o objetivo me parece ser o primeiro contato do aluno com um modelo de classificação de imagens, poderia ser ofertado um "modelo sequencial estruturado" (scaffolding), talvez com botões e abas numeradas e textos explicativos através de hints (textos que aparecem quando o mouse passa sobre eles) dos mesmos, para que o aluno novata entenda o propósito geral do que está sendo realizado ou o que deve ser avaliado em cada etapa (Por exemplo: como identificar se a matriz de confusão é boa?)
Não ter acesso a imagens para poder visualizar os resultados de uma predição sem fazer upload	Acredito que a interface tem muitas opções e acredito que faltam informações de fácil acesso sobre elas, o que é um DataLoader, por que eu preciso disso? Digo isto para o sentido de ensino de ML, para uso de alguém que já tem esse conhecimento, achei ótimo
Não que não tenha gostado, mas levei uns 2 minutos para entender (fiquei lendo o código dentro da primeira célula) que a proposta era utilizar o JupyterNotebooks totalmente de forma visual, apenas pelas abas e botões. O que agora me parece excelente!	Acho que hiperlinks com definições, helps e tutoriais para cada opção a ser escolhida facilitaria a operação e aprendizado do usuário.
Nada a comentar.	-
Nada	
Gostei de tudo :D	
Embora o código esteja disponível, ele não é obrigatório (entendimento ou mesmo	
1 - 14 / 14 < >	1 - 9 / 9 < >

Vulcan: Desenvolvimento de uma Ferramenta Visual de Classificação de Imagens para o Ensino de Machine Learning no Ensino Médio

Augusto C. M. Franz¹, Christiane Gresse Von Wangenheim¹, Jean C. R. Hauck¹, Ramón M. Martín¹

¹Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis / SC, Brasil

a.franz@grad.ufsc.br, c.wangenheim@ufsc.br, jean.hauck@ufsc.br, ramon.mayor@posgrad.ufsc.br

Abstract. *As Machine Learning (ML) is becoming omnipresent in our lives it becomes important to start teaching a basic understanding of ML concepts early to enable the understanding of its potential and limits. Thus, in order to facilitate the learning of the application of basic ML concepts to novices, we developed a visual layer that provides a comprehensive support for all process steps involved in developing image classification models. In order to ease the transition to conventional environments later on the visual layer is implemented with Jupyter Widgets within a Jupyter notebook running on Google Colab. Results of a preliminary evaluation provide a first indication that the visual layer can be a useful, functional, performance-efficient and usable support for teaching and learning the development of image classification models to novices in high school and higher education.*

Keywords: *Machine Learning, Teaching, Novice, Image classification, Visual tool.*

Resumo. *Como Machine Learning (ML) está se tornando onipresente em nossas vidas, torna-se importante começar a ensinar uma compreensão básica dos conceitos de ML desde o início para permitir a compreensão de seu potencial e limites. Assim, a fim de facilitar o aprendizado e aplicar os conceitos básicos de ML para iniciantes, desenvolvemos uma camada visual que fornece um amplo suporte para todas as etapas do processo envolvidas no desenvolvimento de modelos de classificação de imagens. A fim de facilitar a transição para ambientes convencionais mais tarde na camada visual é implementado com Jupyter Widgets dentro de um notebook Jupyter em execução no Google Colab. Os resultados de uma avaliação preliminar fornecem uma primeira indicação de que a camada visual pode ser útil, funcional, eficiente em termos de desempenho e utilizável para ensinar e aprender o desenvolvimento de classificação de modelos de imagens para iniciantes no ensino médio e superior.*

Palavras-chave: *Machine Learning, Ensino para iniciantes, Classificação de imagens, Ferramenta visual*

1. Introdução

Machine Learning (ML) está presente em muitos dispositivos e serviços que fazem parte da nossa vida, como diagnóstico de saúde, detecção de fraude ou veículos autônomos. Assim, preparar jovens para se tornarem usuários responsáveis e conscienciosos e criadores de soluções inteligentes, é importante popularizar uma compreensão básica das tecnologias de ML já no K-12 (Touretzky *et al.*, 2019; Wong *et al.*, 2020; Liu *et al.*, 2020).

Para que os alunos se tornem não apenas consumidores de IA, mas criadores de soluções inteligentes, é preciso o ensino da aplicação desses conceitos, para que obtenham uma compreensão mais profunda (Kahn *et al.*, 2018; Touretzky *et al.*, 2019). A aprendizagem ativa que enfatiza o fazer e as experiências diretas é importante, pois ajuda a tornar o ML transparente, permitindo que alunos construam modelos mentais corretos.

Normalmente, os modelos de ML são desenvolvidos usando linguagens de programação baseadas em texto, como Python, em ambientes de programação como Jupyter Notebooks (jupyter.org) em execução no Google Colab (colab.research.google.com), fornecendo uma forma gratuita, aberta, interativa e colaborativa e ambiente acessível para desenvolver modelos de ML na exploração da nuvem. O ambiente Jupyter / Colab já é amplamente aplicado na educação STEAM (Khine e Areepattamannil, 2019), bem como para o ensino de ML no ensino médio (Mobasher *et al.*, 2019; Technogirlz, 2018; Vachovsky *et al.*, 2016).

No entanto, desenvolver modelos de ML usando uma linguagem de programação textual envolve um entendimento dos conceitos de programação e sua sintaxe (McCracken *et al.*, 2001). Portanto, pode ser benéfico começar a ensinar o desenvolvimento de modelos de ML usando ferramentas visuais para que o aluno se concentre nos aspectos conceituais para resolver o problema em questão (Dove *et al.*, 2017). As linguagens visuais podem melhorar a capacidade de aprendizagem de novatos, ajudando-os a evitar erros, favorecer o reconhecimento em vez de lembrar e fornecer conjuntos de instruções limitados específicos de domínio, reduzindo a carga cognitiva (Çakiroğlu *et al.*, 2018).

Já existem ferramentas que fornecem uma interface visual orientada para o fluxo de trabalho, permitindo executar um processo de ML centrado no ser humano de forma interativa e desenvolver modelos de ML de forma rápida (Gresse von Wangenheim *et al.*, 2021). A maioria dessas ferramentas é voltada principalmente para iniciantes, seguindo os princípios de design propostos por Resnick *et al.* (2005) e foi desenvolvida especificamente como ferramentas visuais parcialmente integradas a ambientes de programação baseados em blocos populares, como Scratch, Snap! Ou App Inventor.

Essas ferramentas normalmente incluem um componente para o desenvolvimento de um modelo de ML e um componente de implantação (Rodríguez-García *et al.*, 2020). No entanto, como a maioria dessas transições ocorrem começando em ambientes visuais que foram desenvolvidos especificamente para fins educacionais, pode ser mais difícil para os alunos progredir para o desenvolvimento de modelos de ML de uma forma mais convencional com Python em cadernos Jupyter. Portanto, a transição de uma interface visual para um ambiente convencional pode ser mais suave e facilitar o processo de aprendizagem.

Neste contexto, apresentamos uma camada visual para notebooks Jupyter fornecendo um suporte baseado em fluxo de trabalho com widgets interativos para o desenvolvimento de modelos de ML para classificação de imagens usando aprendizagem supervisionada no contexto do ensino de computação no ensino médio. Nesse sentido, a seguinte questão de pesquisa é abordada nesta pesquisa: É possível fornecer uma camada visual para notebooks Jupyter com um suporte útil e abrangente para o ensino do desenvolvimento de modelos de ML para iniciantes no ensino médio? Os resultados desta pesquisa podem auxiliar designers e educadores instrucionais a apoiar o ensino de ML no ensino médio, bem como pesquisadores a orientar a evolução e aprimoramento dessas ferramentas.

2. Metodologia de pesquisa

Visando o desenvolvimento de uma interface visual que facilite o aprendizado da aplicação dos conceitos de ML, adotamos uma estratégia de pesquisa multi-método.

Análise do estado da arte. Com o objetivo de fornecer uma visão geral sobre o estado da arte atual nos ambientes visuais de ML, realizamos um estudo de mapeamento sistemático seguindo o procedimento proposto por Petersen *et al.* (2015) e Kitchenham *et al.* (2011). Na fase de definição, foram definidas as questões de pesquisa e o protocolo de revisão, bem como as fontes de dados, estratégia de busca, strings de busca e critérios de inclusão/exclusão. A fase de execução foi realizada com base no protocolo de revisão conduzindo a pesquisa nos repositórios especificados. Os resultados da pesquisa inicial foram analisados quanto à sua relevância aplicando os critérios de inclusão/exclusão. Uma vez identificados os estudos relevantes, foram extraídos os dados relativos às questões de pesquisa. Com base nos dados extraídos, os estudos encontrados foram analisados e os resultados sintetizados. Os resultados detalhados desta revisão são apresentados em Gresse von Wangenheim *et al.* (2021).

Design centrado no ser humano da extensão visual do Jupyter Notebook. Seguindo uma abordagem de design centrado no ser humano (ISO / IEC 9241-210, 2019), realizamos as seguintes atividades:

Caracterização do contexto de uso. No âmbito da análise de contexto, analisou-se o público-alvo e as necessidades para o respectivo nível de ensino. Analisamos e ajustamos o processo de desenvolvimento de ML para fins educacionais nesta fase educacional, identificando o objetivo e as tarefas. Também analisamos o ambiente em termos das características da plataforma de destino (Jupyter Notebook e Google Colab).

Especificação dos requisitos do usuário. Identificamos os requisitos funcionais e não funcionais da extensão visual com base na análise de contexto.

Desenvolvimento da camada visual. Uma das alternativas técnicas viáveis para o desenvolvimento de uma interface visual para o notebook Jupyter no Google Colab, adotamos uma abordagem de desenvolvimento de software iterativa e incremental (Larman e Basili, 2003). Nós, iterativa e incrementalmente, desenvolvemos o suporte de software especificando primeiro o conteúdo do design da interface do usuário e, em seguida, desenvolvendo protótipos da interface do usuário que foram concluídos de forma incremental com base no feedback informal de especialistas em ML até cobrir todo o processo de desenvolvimento de ML. Em seguida, foram realizados testes de sistema e com base nos resultados, as correções e melhorias necessárias foram implementadas e testadas.

Avaliação da camada visual. Visando uma avaliação preliminar da camada visual, realizamos um painel de especialistas. Seguindo Wohlin *et al.* (2012) e usando GQM (Basili *et al.*, 1994) definimos sistematicamente o objetivo da avaliação, questões e métricas e elaboramos um questionário para operacionalizar a coleta de dados. Durante a execução, os participantes testaram a camada visual proposta e ao final responderam o questionário. Os dados coletados foram analisados por meio de estatística descritiva e os resultados foram interpretados e discutidos.

3. Visual Layer Classification (Vulcan) para classificação de imagens

O objetivo desta pesquisa é desenvolver uma interface visual baseada na web que permita a estudantes do ensino médio desenvolver modelos de classificação de imagens em notebook Jupyter rodando no Google Colab sem codificação. Ele deve ser usado como uma parte introdutória da educação em ML, preparando e facilitando também uma transição posterior para a programação baseada em texto em Python no mesmo ambiente convencional.

3.1 Análise de contexto

Os usuários-alvo são alunos do ensino médio no Brasil de cerca de 15 a 18 anos. De acordo com as diretrizes curriculares nacionais, os alunos já adquiriram competências básicas em matemática, línguas, ciências, etc. No entanto, normalmente o conhecimento em línguas estrangeiras como o inglês pode ser limitado, exigindo uma interface de usuário na língua nativa (português do Brasil). Na medida em que nem a computação nem a Inteligência Artificial são uma parte explícita do currículo escolar nacional, a maioria dos alunos não tem conhecimento prévio sobre programação ou Aprendizado de Máquina.

Com base nas diretrizes curriculares de IA (Touretzky *et al.*, 2019; Long e Magerko, 2020), nos concentramos, como parte desta pesquisa, nos objetivos de aprendizagem que visam permitir que os alunos apliquem conceitos básicos de ML relacionados ao desenvolvimento de modelos de aprendizado profundo. Nos concentramos na tarefa específica de classificação de imagens, normalmente ministrada como uma das primeiras aplicações (Marques *et al.*, 2020). Conforme apresentado na Figura 1, o processo de ML é derivado de processos de ML interativos comumente propostos (Amazon 2019; Amershi *et al.* 2019; Fiebrink e Gillies, 2018; Mathewson 2019; Watanabe *et al.* 2019) ajustados e simplificados para manter a carga cognitiva apropriada para iniciantes em ML. Além disso, para facilitar o desenvolvimento de um modelo de ML com base em um pequeno conjunto de dados, adotamos uma abordagem de aprendizagem por transferência. Com o aprendizado de transferência, os usuários podem adicionar seus próprios dados e treinar novamente um modelo em cima de um modelo de base previamente treinado que aprendeu um domínio específico de um grande conjunto de dados, beneficiando-se da necessidade de menos dados e tempo de treinamento para criar modelos úteis e precisos.



Figura 1. Processo de desenvolvimento de ML interativo (Gresse von Wangenheim e von Wangenheim, 2021)

3.2 Requisitos

O principal requisito funcional da camada visual é fornecer suporte para que o aluno desenvolva de forma interativa um modelo de ML para classificação de imagens no Jupyter Notebook/Google Colab. Portanto, ele precisa oferecer suporte a todo o processo de desenvolvimento de ML, conforme ilustrado na Figura 1: análise de requisitos, preparação de dados, treinamento de modelo, avaliação de modelo, predição e exportação. A implantação do modelo, conforme ocorre nas plataformas de destino, está fora do escopo aqui. Portanto, objetivamos o desenvolvimento de uma camada visual baseada em fluxo de trabalho visual para o notebook Jupyter.

Para possibilitar o uso do suporte online e o uso gratuito de recursos computacionais via Google Colab para treinamento do modelo de ML, ele precisa estar totalmente integrado aos notebooks jupyter e poder rodar no Google Colab. A ferramenta também deve permitir a conexão com o Google Drive para carregar facilmente o conjunto de dados previamente separado em pastas para cada categoria. Além disso, durante a etapa de previsão, também deve ser possível fazer o upload de uma nova imagem para realizar a previsão. Imagens de diferentes tamanhos e formatos devem ser aceitas e tratadas para se tornarem uniformes.

Outros requisitos não funcionais incluem:

- Design visual: seguindo a identidade visual da iniciativa Computação na Escola
- Usabilidade: Eficácia: 90% dos alunos conseguem desenvolver um modelo de ML, Eficiência: em menos de 1 hora, com satisfação mínima de 75 pontos da escala do SUS
- Extensibilidade: o sistema deve permitir que novos critérios de análise sejam adicionados ao sistema no futuro.
- Expressões idiomáticas: A camada visual deve estar disponível em português do Brasil, pois nesta fase educacional muitos alunos ainda não são fluentes em inglês, além do inglês, de forma a possibilitar uma aplicação em maior escala.

3.3 Arquitetura do VULCAN

A implementação da camada visual consiste em um ambiente que roda em um notebook Jupyter dentro do Google Colab (Figura 2). Os dados são carregados do Google Drive para gerenciar, visualizar e compartilhar o conjunto de dados facilmente. O modelo também é treinado no ambiente Google Colab.

Para a implementação dos elementos visuais, a biblioteca Jupyter Widgets (Ipywidgets, 2021) é utilizada como principal referência para os elementos visuais utilizados no Jupyter. Para a tarefa de classificação de imagens, usamos a estrutura de *deep learning* fast.ai, sendo uma estrutura de alto nível com uma das curvas de aprendizado mais rápidas em comparação com outras estruturas (Howards e Gugger, 2020). Apesar de suportar elementos visuais, o Colab não foi projetado como uma interface gráfica ou um aplicativo de célula. Normalmente, várias células são usadas, uma com cada pedaço de código referindo-se à respectiva etapa do processo de ML. Porém, também é possível encapsular todos os processos de IA/ML de acordo com os elementos visuais disponíveis e realizar a aplicação ponta a ponta em apenas uma célula.

A distribuição do Visual Layer Classification é feita por meio de um pacote PIP para disponibilizá-lo também no notebook Jupyter com o mínimo de linhas de código, ocultando os detalhes de implementação dos alunos.

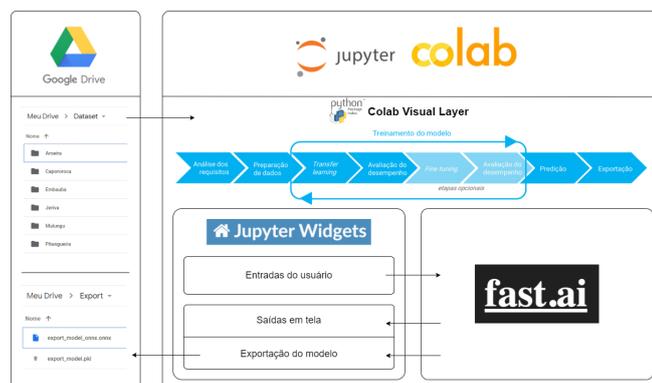


Figura 2. Visão geral do Vulcan

Toda a camada foi desenvolvida em Python, pois roda nativamente no colab e tem compatibilidade total com os frameworks selecionados.

De acordo com a análise de requisitos e a arquitetura definida seguindo o processo de ML interativo centrado no ser humano (Gresse von Wangenheim e von Wangenheim, 2021) as necessidades de entrada e saída de dados foram especificadas, especificando o suporte pela *Classificação da Camada Visual* (Vulcan) conforme apresentado em Tabela 1.

Tabela 1. Especificação e visualização do suporte fornecido

Processo de ML	Etapa	Especificação de funcionalidade e parâmetros
Análise de Requisitos	Especificação da(s) tarefa(s) e requisitos do sistema de ML	Especificação da tarefa, tipo de tarefa, categorias, fonte (s) de dados, quantidade esperada de dados necessários, descrição do formato de dados esperado, rotulagem de dados, níveis de desempenho esperados.
Preparação de dados	Definição do conjunto de dados no Google Drive	Especificação do diretório no Google Drive onde o conjunto de dados da imagem está disponível com um subdiretório para cada categoria.
	Visualize a distribuição do conjunto de dados	Um histograma mostrando as quantidades de imagens para cada categoria no conjunto de dados.
	Padronizar imagens: redimensionamento	Por padrão, todas as imagens são redimensionadas para um tamanho padrão (224px usando "squish") ou o usuário pode indicar manualmente o tamanho e também o método de redimensionamento ("crop", "pad", "squish").
	Padronizar imagens: transformações e aumentações	Por padrão, as transformações e aumentos são desabilitados. O usuário pode selecionar manualmente os métodos apropriados de acordo com o domínio de aplicação específico e o tipo de imagem, incluindo: espelho horizontal, espelho vertical, zoom máximo, zoom mínimo, iluminação máxima, intensidade de transformações, rotação máxima, intensidade de luz, inclinação.
	Conjunto de dados de treinamento e validação	O usuário pode especificar a porcentagem de imagens do conjunto de dados usado para treinamento e validação. Por padrão, 20% das imagens são usadas para validação.
	Tamanho do lote	O usuário pode especificar o tamanho do lote, que por padrão é especificado como 16.
	Criar dataloader	O dataloader é criado de acordo com os parâmetros definidos anteriormente no conjunto de dados (redimensionamento, transformações e aumentações (se houver), e definições do lote). Ele também converte todas as imagens do conjunto de dados para o mesmo formato (jpg) e padroniza os nomes de acordo com a pasta.
	Visualização do lote	O lote criado pode ser visualizado especificando-se o número de imagens a serem mostradas e se apenas variações da mesma imagem devem ser mostradas.
Aprendizagem por transferência	Selecione a arquitetura	O usuário pode selecionar qualquer uma das arquiteturas padrão cobertas pelo fastai (por exemplo, alexnet, resnet18, resnet34, resnet50, resnet101, ...). Por padrão, um resnet18 é selecionado.
	Especifique métricas	O usuário pode selecionar as métricas a serem usadas durante o aprendizado de transferência (taxa de erro ou precisão).
	Criar learner	O learner é criado de acordo com os parâmetros definidos anteriormente.
	Especifique o número de épocas	O usuário pode especificar o número de épocas que variam de 1 a 50.
	Treinar modelo	A aprendizagem de transferência é executada e os resultados são exibidos.
	Limpeza de dados	Visualiza as imagens no conjunto de dados e as classificações feitas e permite ao usuário alterar rótulos e/ou excluir imagens do conjunto de dados.
Avaliação de aprendizagem de transferência	Analisar a matriz de confusão	Exibe a matriz de confusão como resultado da aprendizagem por transferência.
	Análise exatidão, precisão, recall, pontuação F1	Exibe indicadores de desempenho, incluindo precisão total, bem como precisão, recall e pontuação F1 por categoria.
	Visualize as principais perdas	Exibe as perdas principais (por padrão, as 3 perdas principais).
	Listar itens mais confundidos	Lista as imagens mais confundidas (por padrão, as 3 mais confundidas).

Fine tuning (opcional)	Descongelo modelo	Descongela o modelo como parte do ajuste fino.
	Encontre a melhor taxa de aprendizagem	Procure a melhor taxa de aprendizagem para o modelo e salve este valor com a possibilidade de usar métodos diferentes, incluindo min, step, valley e slide (usando vale por padrão).
	Especifique o número de épocas	O usuário pode especificar o número de épocas que variam de 1 a 50.
	Treinar modelo otimizado	O <i>learner</i> é executado e os resultados são exibidos.
Avaliação de ajuste fino	Analisar a matriz de confusão	Exibe a matriz de confusão do modelo treinado.
	Análise exatidão, precisão, recall, pontuação F1	Exibe a precisão por categoria e outros modelos de métricas de desempenho de treinamento como pontuação f1.
	Visualize as principais perdas	Exibe as perdas principais (por padrão, as 3 perdas principais).
	Lista para mais confundidos	Lista as imagens mais confundidos (por padrão, as 3 mais confundidos).
Predição	Faça upload de uma nova imagem não vista	Permite fazer upload de uma imagem (por vez).
	Classificação de predição	Exibe a imagem e o resultado da predição indicando a classificação (categoria).
Exportar	Exportar modelo PKL	Exporta o modelo para o diretório fornecido usando pickle_protocol.
	Exportar modelo ONNX	Exporta o modelo para o diretório fornecido no formato ONNX (Open Neural Network Exchange), permitindo a interoperabilidade entre os principais frameworks.

O pacote *vulcan-colab* está disponível na biblioteca PyPI pode ser importado diretamente para qualquer *notebook* Colab. Todo o código desenvolvido está disponível no repositório de códigos UFSC (<https://codigos.ufsc.br/gqs/colabvisuallayer>) sob a licença BSD 3-CLAUSE. Para orientar o uso da camada visual foi criada a documentação sobre cada uma das etapas suportadas pela camada visual (<https://vulcan-colab.readthedocs.io/>).

4. Avaliação

Para avaliar a qualidade da solução, realizamos uma avaliação preliminar na forma de um painel de especialistas. Nosso objetivo é analisar a qualidade da camada visual em termos de utilidade, utilidade, desempenho e usabilidade do ponto de vista de professores e alunos no contexto da educação em computação. De acordo com nosso foco no nível médio, incluímos professores do ensino médio. No entanto, devido ao fato de que atualmente a maioria dos alunos do ensino médio não possui nenhum conhecimento sobre ML e, portanto, podem não ser capazes de avaliar tal suporte, incluímos neste estudo preliminar alunos de graduação com conhecimento sobre ML. E, tendo em vista que tal interface visual também pode ser utilizada em um momento inicial nos cursos de ML no ensino superior, ampliou o grupo de participantes incluindo também professores do ensino médio.

A avaliação da ferramenta foi realizada em agosto de 2021. Inicialmente foram convidados 26 alunos de graduação e pós-graduação e professores do ensino médio e ensino superior. Os participantes foram selecionados por amostragem de conveniência junto a membros da iniciativa Computação na Escola/INCoD/INE/UFSC com conhecimento prévio em ML para classificação de imagens. Os participantes foram convidados por e-mail explicando o objetivo da avaliação e garantindo o sigilo. A participação foi voluntária.

Ao final foram obtidas 14 respostas completas, representando uma taxa de resposta de 54%, alguns assumindo mais de um papel como professor do ensino médio e aluno da pós-graduação, por exemplo.

Figura 3 apresenta a distribuição demográfica dos participantes da avaliação. Mais detalhes referentes aos dados coletados são apresentados no apêndice A e também disponível online.

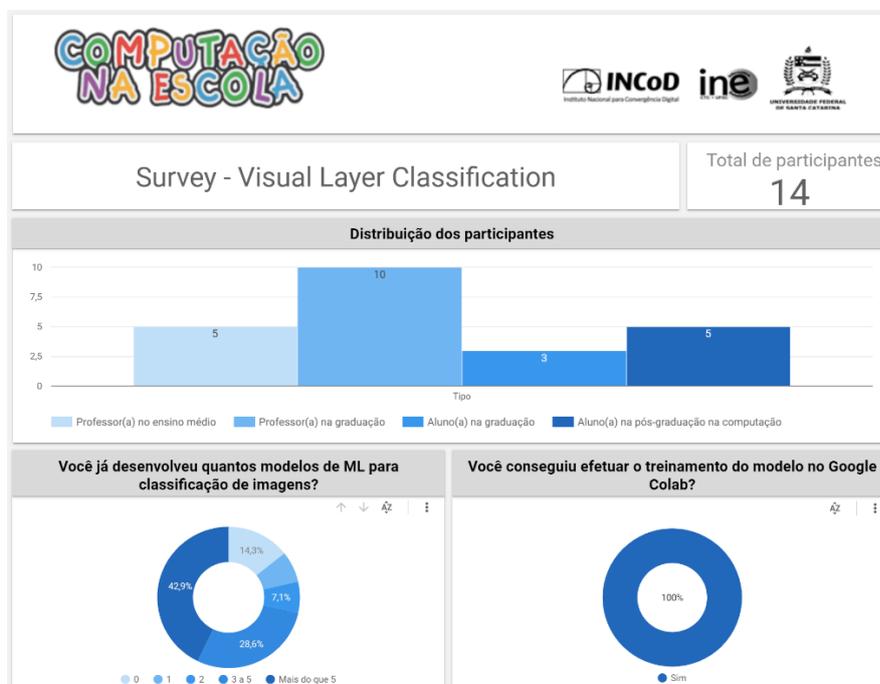


Figura 3 - Visão geral da distribuição demográfica dos participantes.

4.1. Análise dos resultados

Utilidade. Todos os alunos participantes acharam a ferramenta útil para aprender ML, e 9 entre 10 professores acharam a ferramenta útil para ensinar ML no ensino médio. Quando perguntados para ensinar alunos iniciantes na graduação todos os professores acharam a ferramenta útil. A maioria dos participantes acha que pode ajudar a aprender conceitos de ML, além de achar que a ferramenta pode ser mais útil para aprender a desenvolver um modelo de ML em comparação ao ambiente tradicional, com exceção de um aluno (Tabela 2).

Tabela 2 - Análise da utilidade da ferramenta.

Papal	Questão de análise	Discordo totalmente		Discordo		Não concordo nem discordo		Concordo		Concordo totalmente		Total
		0	0%	1	10%	0	0%	5	50%	4	40%	
Professor	Acho que a ferramenta é útil para ensinar ML no ensino médio	0	0%	1	10%	0	0%	5	50%	4	40%	10
Aluno	A ferramenta é útil para aprender ML	0	0%	0	0%	0	0%	5	55,6%	4	44,4%	9
Professor	Acho que a ferramenta é útil para ensinar ML para iniciantes na graduação / alunos na graduação	0	0%	0	0,00%	0	0%	5	50%	5	50%	10
Professor	Acho que pode ajudar os alunos a aprender conceitos de ML	0	0%	0	0,00%	2	20%	3	30%	5	50%	10
Aluno	Acho que pode me ajudar a aprender conceitos conceituais de ML	0	0%	0	0,00%	1	11,1%	6	66,6%	2	22,2%	9

Professor	Eu acho que pode ser mais útil do que a interface de programação baseada em texto convencional do Jupyter para começar a ensinar o desenvolvimento de um modelo de ML para iniciantes.	0	0%	0	0,00%	1	10%	4	40%	5	50%	10
Aluno	Eu preferiria começar a aprender o desenvolvimento de um modelo de ML usando essa interface visual do que usando a interface de programação baseada em texto convencional do Jupyter	0	0%	1	11,11%	2	22,2%	2	22,2%	4	44,4%	9
	Total	0	0%	2	2,9%	6	8,9%	30	44,7%	29	43,2%	67

Funcionalidade. Em geral, a completude e a corretude obtiveram boas avaliações. Apenas dois participantes apontaram partes não suportadas pela ferramenta, sendo ela a rotulação das imagens. Como definido isso ocorre dentro da ferramenta, mas houve uma falha na comunicação com os participantes e acabando não deixando isso explícito. Mas estes relatos mostram que o processo de rotulação pode ser retrabalhado, e quem sabe esteja dentro da ferramenta em trabalhos futuros. Em relação a erros, poucos usuários relataram problemas, entre eles a probabilidade da predição que muitas vezes acertava a classe mas apresentava um percentual incorreto.

Tabela 3 - Análise da funcionalidade da ferramenta.

Pergunta	sim		não		total
	count	percentage	count	percentage	
Existem etapas/partes para ensinar o desenvolvimento da classificação de imagens para iniciantes que não são suportadas pela ferramenta?	12	85,71%	2	14,29%	14
Você notou algum erro em relação à funcionalidade da ferramenta?	3	21,43%	11	78,57%	14

Desempenho. Todos os participantes acharam o resultado da ferramenta satisfatório, conseguindo executar todas as etapas rapidamente, com exceção das parametrizações que naturalmente aumentam o tempo de execução do treinamento do modelo.

Tabela 4 - Análise do desempenho da ferramenta.

Pergunta	sim		não		total
	count	percentage	count	percentage	
O desempenho da ferramenta é satisfatório?	14	100%	0	0%	14

Usabilidade. Em geral, a usabilidade da ferramenta teve um *feedback* muito positivo. Em relação a eficácia 13 entre os 14 participantes conseguiram desenvolver o modelo para classificação de imagens.

Tabela 5 - Análise da usabilidade da ferramenta.

Pergunta	sim		não		total
	count	percentage	count	percentage	
Você conseguiu desenvolver um modelo de ML para classificação de imagens com a ferramenta?	13	92,86%	1	7,14%	14
Você considera apropriado o tempo que levou para desenvolver o modelo de ML com a ferramenta?	14	100%	0	0%	14

A ferramenta também se mostrou eficiente, como todos os usuários considerando apropriado o tempo que levou para desenvolver o modelo. Além disso, 11 dos 13 usuários que concluíram o desenvolvimento do modelo terminaram a tarefa em até 30 minutos. Também 10 dos 14 participantes acham que o tempo levado para realizar todo o processo é mais rápido do que em um ambiente baseado em texto.

Tabela 6 - Análise do tempo de treinamento da ferramenta.

Pergunta	Até 30 min		Até 60 min		total
Quanto tempo aproximadamente você demorou para desenvolver um modelo de ML usando a ferramenta?	11	84,62%	2	15,38%	13

Tabela 7 - Análise da comparação da ferramenta com ambiente tradicional.

Pergunta	menos tempo		mesmo ou mais tempo		total
Em comparação com uma versão de programação baseada em texto de um Jupyter notebook, você demorou menos tempo, mais tempo ou o mesmo para desenvolver um ML para classificação de imagens?	10	71,43%	4	28,57%	14

Para medir a satisfação dos usuários foi aplicado o questionário SUS, e como resultado foi obtida a boa média de 80,3%. Individualmente a nota mais baixa foi de 72,5%, ainda indicando uma boa satisfação. A questão individual com a média mais baixa foi a que foi perguntado se é possível aprender como usar esse sistema rapidamente. A maior pontuação de uma questão individual foi referente a consistência do sistema, com nenhum participante indicando inconsistências.

Tabela 8 - Análise do questionário SUS.

Pergunta	Discordo totalmente	Discordo	Não concordo nem discordo	Concordo	Concordo totalmente	Pontuação SUS
Eu acho que gostaria de usar esse sistema com frequência.	0 (0%)	0 (0%)	3 (21,4%)	7 (50%)	4 (28,6%)	7,68
Eu acho o sistema desnecessariamente complexo.	5 (35,7%)	8 (57,1%)	1 (7,1%)	0 (0%)	0 (0%)	8,21
Eu achei o sistema fácil de usar.	0 (0%)	0 (0%)	0 (0%)	10 (71,4%)	4 (28,6%)	8,21
Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	5 (35,7%)	6 (42,9%)	2 (14,3%)	1 (7,1%)	0 (0%)	7,68
Eu acho que as várias funções do sistema estão muito bem integradas.	0 (0%)	0 (0%)	2 (14,3%)	7 (50%)	5 (35,7%)	8,04
Eu acho que o sistema apresenta muita inconsistência.	9 (64,3%)	5 (35,7%)	0 (0%)	0 (0%)	0 (0%)	9,11
Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.	0 (0%)	2 (14,3%)	2 (14,3%)	6 (42,9%)	4 (28,6%)	7,14
Eu achei o sistema atrapalhado de usar.	4 (28,6%)	9 (64,3%)	1 (7,1%)	0 (0%)	0 (0%)	8,04
Eu me senti confiante ao usar o sistema.	0 (0%)	0 (0%)	3 (21,4%)	6 (42,9%)	5 (35,7%)	7,86
Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.	6 (42,9%)	7 (50%)	1 (7,1%)	0 (0%)	0 (0%)	8,39
Total SUS						80,36

Em relação a operabilidade ficou evidente que apesar de os participantes conseguirem executar o processo em geral faltavam informações sobre como fazer, o que exatamente estavam fazendo ou como interpretar as entradas de dados e retornos.

Tabela 9 - Análise da operabilidade da ferramenta.

pergunta	sim		não	
Você acha que a ferramenta de avaliação possui elementos ambíguos ou difíceis de entender?	5	35,71%	9	64,29%

4.2. Discussão

Pontos fortes. O *feedback* dos participantes fornece uma primeira indicação que o *Visual Layer Classification* pode ser útil, funcional, eficiente em desempenho e ter boa usabilidade. Como pontos fortes, os avaliadores destacaram, principalmente, a simplicidade e facilidade de desenvolver um modelo de ML de ponta a ponta sem necessariamente ter experiência com códigos e também sua completude. Os participantes elogiaram a praticidade do fluxo do processamento, podendo contemplar etapas que são mais complicadas ou adicionais caso desejem ou não, como realizar as aumentações com transformações, definir como quais métricas serão utilizadas e acompanhadas bem além das várias opções de visualização dos resultados. A possibilidade de poder realizar muitos testes variando as opções disponíveis também foi elogiada, uma vez que se pode demonstrar bem o ciclo de *use-modify-create*.

Sugestões de melhoria. As principais sugestões de melhoria foram principalmente relacionadas à falta de informações extras e explicações conceituais em geral sobre o fluxo, entradas e saídas. Parte disso já foi resolvido com a implementação da documentação externa contendo mais detalhes sobre cada etapa do processo em geral. Cada seção agora tem um ponto de interrogação com um link direto a seção da documentação referente aquela parte do processo, cobrindo todos os elementos da tela. Outro detalhe mencionado foi a respeito dos elementos visuais que poderiam ser melhorados, ficando mais polidos ou apresentando barras de progresso para facilitar o andamento de determinados processos. Além disso, foi sugerida a possibilidade de mais interação dos alunos com o código.

Ameaças a validade da avaliação. 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. Embora os participantes foram 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 conhecimento prévio dos participantes. Assim, indicando a necessidade de estudos futuros com maior número de participantes do perfil específicos do público alvo. Também é importante ressaltar que a ferramenta ainda estava em desenvolvimento quando foi submetida ao painel de especialistas, logo podendo apresentar erros e inconsistências durante os testes.

Além disso, o tamanho da amostra pode comprometer a generalização dos resultados. O estudo foi baseado em um total de 14 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).

6. Conclusão

Neste artigo apresentamos Vulcan - uma camada visual com o objetivo de facilitar a aprendizagem de conceitos de ML voltada principalmente para alunos do ensino médio. Pretende-se que seja adotado no início dos cursos de ML para iniciantes com o objetivo de reduzir a carga cognitiva. Com base nas diretrizes curriculares de IA (Touretsky et al., 2019; Long e Magerko, 2020), nos concentramos, como parte desta pesquisa, em objetivos de aprendizagem com o objetivo de permitir que os alunos apliquem conceitos básicos de ML relacionados ao desenvolvimento de modelos de Deep Learning. Centramo-nos no apoio à tarefa específica de classificação de imagens, normalmente ministrada como uma das primeiras aplicações (Marques et al., 2020). Para facilitar a transição para ambientes convencionais posteriormente, ele é implementado em um notebook Jupyter rodando no Google Colab. A camada visual também fornece suporte para um processo de desenvolvimento de ML completo, incluindo análise de requisitos, preparação de dados, treinamento de modelo, avaliação e exportação para implantação (Amazon 2019; Amershi et al. 2019; Mathewson 2019; Watanabe et al. 2019, Fiebrink e Gillies, 2018) ajustada e simplificada para manter a carga cognitiva adequada para iniciantes em ML.

A camada visual tem usado a biblioteca Jupyter Widgets (Ipywidgets, 2021) e o framework de aprendizagem profunda fast.ai (Howards e Gugger, 2020). A camada visual está sendo distribuída como um pacote PIP para torná-lo disponível no notebook Jupyter com o mínimo de linhas de código, ocultando os detalhes de implementação dos alunos. Dessa forma, ele fornece uma interface de entrada visual, generalizada e flexível para fazer modelos de classificação de ML que podem ser facilmente desenvolvidos sem experiência anterior ou conhecimento em ML ou codificação. A fim de facilitar sua aplicação em sala de aula e ilustrar seu uso, também preparamos um modelo de ML exemplar para a classificação de espécies de árvores, incluindo também um conjunto de dados preparado.

Os resultados da avaliação fornecem uma primeira indicação de que a camada visual pode ser um suporte útil, funcional, com desempenho eficiente e utilizável para ensinar o desenvolvimento de modelos de classificação de imagens para iniciantes no ensino médio e superior. Um dos principais pontos fortes da camada visual é a indicação incorporada das etapas e parâmetros necessários, bem como os valores padrão típicos que facilitam a experimentação do desenvolvimento de modelos de ML por novatos.

Com base nas oportunidades de melhoria identificadas, estamos atualmente desenvolvendo documentação adicional e um curso para fornecer mais explicações sobre como usar a camada visual, bem como a terminologia e os conceitos de ML usados. Outras oportunidades de melhoria relacionadas à usabilidade da camada visual incluem a inclusão de mais feedback visual. Com base no feedback inicial promissor, também estamos planejando a aplicação e avaliação da ferramenta em sala de aula.

Referencias

Amazon. (2019). *Amazon Machine Learning*, AWS Documentation. <https://docs.aws.amazon.com/machine-learning/latest/dg/building-machine-learning.html>

Armoni, M., Meerbaum-Salant, O., Ben-Ari, M. (2015). From Scratch to “Real” Programming. *ACM Trans. Comput. Educ.* 14, 4, Article 25, 15 pages. DOI:<https://doi.org/10.1145/2677087>

Barba, L. A. (2019) Teaching and Learning with Jupyter. GitHub, Creative Commons Attribution CC-BY 4.0 International license, URL <https://jupyter4edu.github.io/jupyter-edu-book/>

Beg, M. *et al.*, (2021) "Using Jupyter for Reproducible Scientific Workflows" in *Computing in Science & Engineering*, vol. 23, no. 02, pp. 36-46, 2021. doi: 10.1109/MCSE.2021.3052101

Bhatia, N. (2020). Using Transfer Learning, Spectrogram Audio Classification, and MIT App Inventor to Facilitate Machine Learning Understanding, M.Eng thesis, Elect. Eng. Comput. Sci., Massachusetts Inst. of Technol., Cambridge, 2020.

Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.), *Usability Evaluation in Industry*. London: Taylor and Francis.

Carney, M. *et al.* (2020). Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification. In *Proceedings of Conference on Human Factors in Computing Systems*, ACM, DOI:10.1145/3334480.3382839.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319–340, doi:10.2307/249008

Dove, G., Halskov, K., Forlizzi, J., & Zimmerman, J. (2017). UX design innovation: Challenges for working with machine learning as a design material. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (pp. 278-288). New York, NY: ACM.

Dudley, J. J., Kristensson, P. O. (2018). A Review of User Interface Design for Interactive Machine Learning. *ACM Trans. Interact. Intell. Syst.*, 8(8), 2018. doi: 10.1145/3185517

Fiebrink, R., Gillies, M. (2018). Introduction to the Special Issue on Human-Centered Machine Learning. *ACM Trans. Interact. Intell. Syst.* 8, 2, Article 7 (July 2018), 7 pages. DOI:<https://doi.org/10.1145/3205942>

Frederickson, B. (2019) Ranking Programming Languages by GitHub Users. URL <https://www.benfrederickson.com/ranking-programming-languages-by-github-users/>

Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410–8415. <https://doi.org/10.1073/pnas.1319030111>

Gillies, M. *et al.* (2016) Human-Centered Machine Learning. *Conference on Human Factors in Computing Systems (CHI 2016)*, ACM, May 2016, San Jose, California, USA, 3558-3565, <https://hal.inria.fr/hal-01437057/document>

Goodfellow, I., Bengio, Y., Courville, A. (2016). *Deep Learning*. Cambridge: MIT Press.

Google. (2017). *Teachable Machine*. <https://experiments.withgoogle.com/teachable-machine>

Gresse von Wangenheim, C., Hauck, J. C. R., Pacheco, F. S., Bertoneceli Bueno, M. F. (2021). Visual Tools for Teaching Machine Learning in K-12: A Ten-Year Systematic Mapping. *Education and Information Technologies*, <https://doi.org/10.1007/s10639-021-10570-8>

C. Gresse von Wangenheim, C. Gresse von Wangenheim. Overview on a human-centric interactive ML process for teaching ML in K-12. Working Paper WP_GQS_01_2021_v10, GQS/INCoD/UFSC, Florianópolis, Brazil, 2021.

Hakim, C. (1987). *Research Design: Strategies and Choices in the Design of Social Research*, London: Allen & Unwing.

He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proc. of the Conference on Computer Vision and Pattern Recognition*, IEEE, 770-778, doi: 10.1109/CVPR.2016.90.

ISO 9241-210:2019 Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems

ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models

Hayes, B. (2019). *Most Popular Machine Learning Frameworks and Products Used by Data Professionals*. <http://businessoverbroadway.com/2019/01/27/most-popular-machine-learning-frameworks-and-products-used-by-data-professionals/>

J. Howards, S. Gugger. *Deep Learning for Coders with Fastai and Pytorch: AI Applications Without a PhD*. O'Reilly Media, 2020.

Khine, M. S. , Areepattamannil, S. (2019). *STEAM Education: Theory and Practice*. Springer, Cham.

Kluyver, T. et al. (2016). Jupyter Notebooks-a publishing format for reproducible computational workflows. *Proc. of the 20th International Conference on Electronic Publishing.*, IosPress, 87 – 90.

Krizhevsky, A., Sutskever, I., Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM* 60(6), 84–90. DOI:<https://doi.org/10.1145/3065386>

Lao, N. (2020). *Reorienting Machine Learning Education Towards Tinkerers and ML-Engaged Citizens*. PhD dissertation, Elect. Eng. Comput. Sci., Massachusetts Inst. of Technol., Cambridge, 2020.

Lao, N., Bhatia, N. (2020). Using Transfer Learning, Spectrogram Audio Classification, and MIT App Inventor to Facilitate Machine Learning Understanding. *Proc. of the 4th International Conference on Computational Thinking Education*, online.

Lawshe, C. H. (1975). A quantitative approach to content validity. *Personnel psychology*, 28(4), 563–575.

LeCun, Y., Kavukcuoglu, K., Farabet, C. (2010). Convolutional networks and applications in vision. *Proc. of the International Symposium on Circuits and Systems*, IEEE, 2010, 253-256, doi: 10.1109/ISCAS.2010.5537907.

Liu et al. (2020). Emerging and Changing Tasks in the Development Process for Machine Learning Systems. Proc. of the IEEE/ACM International Conference on Software and System Processes, Seoul, Republic of Korea.

Lye, S. Y., 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, 51-61.

Lytle, N., Cateté, V., Boulden, D., Dong, Y., Houchins, J., Milliken, A., Isvik, A., Bounajim, D., Wiebe, E., Barnes, T. (2019). Use, modify, create: Comparing computational thinking lesson progressions for stem classes. Proc. of the *ACM Conference on Innovation and Technology in Computer Science Education*, 395–401.

Makeblock. *Make with Code*. (2019). <https://www.mblock.cc/en-us/>

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

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Education, New York.

Mobasher, B. et al. (2019). Data Science Summer Academy for Chicago Public School Students. *ACM SIGKDD Explorations Newsletter*, 21(1)

Jeffrey M. Perkel. Data visualization tools drive interactivity and reproducibility in online publishing. *Nature*, **554**, 133-134 (2018) doi: <https://doi.org/10.1038/d41586-018-01322-9>

M. Perkel, J. M.. Why Jupyter is data scientists' computational notebook of choice. *Nature*, 563, 145-146 (2018) <https://doi.org/10.1038/d41586-018-07196-1>

Patel, K., Fogarty, J., Landay, J. A., Harrison, B. (2008). Investigating statistical machine learning as a tool for software development. Proc. of the 26th annual SIGCHI Conference on Human factors in computing systems. ACM, New York, NY, USA, 667–676. DOI: <http://dx.doi.org/10.1145/1357054.1357160>

Petersen, K. et al. (2008) Systematic mapping studies in software engineering. *Proc. of the 12th International Conference on Evaluation and Assessment in Software Engineering*, Bari, Italy, 68–77.

Piazzentin Ono, Freire, J., Silva, C. T. (2021). Interactive Data Visualization in Jupyter Notebooks. *Computing in Science & Engineering*, 23(2), 99-106, doi: 10.1109/MCSE.2021.3052619.

Project Jupyter, Bussonnier M, Forde J, Freeman J, Granger B, Head T, Holdgraf C, Kelley K, Nalvarte G, Osheroff A, Pacer M, Panda Y, Perez F, Ragan-Kelley B, Willing C (2018) Binder 2.0 - Reproducible, interactive, sharable environments for science at scale. Proc. of the 17th Python in Science Conference, Austin, TX, USA, pp 113–120.

Ramos, G., Meek, C., Simard, P., Suh, J., Ghorashi, S. (2020). Interactive machine teaching: a human-centered approach to building machine-learned models, *Human-Computer Interaction*, 35(5-6). https://www.microsoft.com/en-us/research/uploads/prod/2020/05/Interactive_Machine_Teaching_Free_access_.pdf

Resnick, M., Berg, R., Eisenberg, M. (2000). Beyond black boxes: Bringing transparency and aesthetics back to scientific investigation. *The Journal of the Learning Sciences*, 9(1), 7–30.

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). DOI:10.6018/red.410121

Sun, L., Zhou, Z., Wu, W., Zhang, Y. Zhang, R., & Xiang, W. (2020). Developing a toolkit for prototyping machine learning-empowered products: The design and evaluation of ML-Rapid. *International Journal of Design*, 14(2), 35-50.

Tang, D., Utsumi, Y., Lao, N. (2019). PIC: A Personal Image Classification Webtool for High School Students. *Proc. of the IJCAI EduAI Workshop*, Macao, China.

TechGirlz, 2018. <https://www.techgirlz.org/topic/artificial-intelligence-computers-learn/>

Tissenbaum, M., Sheldon, J., and Abelson, H. (2019). From computational thinking to computational action. *Communications of the ACM*, 62(3), 34–36. DOI:<https://doi.org/10.1145/3265747>

Touretzky, D. S., Gardner-McCune, C., Martin, F., Seehorn, D. (2019). Envisioning AI for K-12: What Should Every Child Know about AI?. *Proc. of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA.

Vachovsky, M. E. et al. (2016). Toward More Gender Diversity in CS through an Artificial Intelligence Summer Program for High School Girls. *Proc. of the 47th ACM Technical Symposium on Computing Science Education*, Memphis, TN, USA.

Wagner, A., Gray, J., Corley, J., Wolber, D. (2013). Using app inventor in a K- 12 summer camp. *Proc. of the 44th ACM Technical Symposium on Computer science education*. ACM., pp. 621–626.

Weintrop, D., Wilensky, U. (2019). Transitioning from introductory block-based and text-based environments to professional programming languages in highschool computer science classrooms. *Computers & Education*, 142, 103646.