ISSN: 2594-8083

(X) Graduação ( ) Pós-Graduação

FERRAMENTAS DE INTELIGÊNCIA ARTIFICIAL: um relato de prática do Centro Internacional de Tecnologia de Software

Bruno de Oliveira Maciel
CITS
bruno.maciel@cits.br

Isabela Dambiski Gomes de Carvalho
CITS
isabela.carvalho@cits.br

Loraine de Fatima Mendes
CITS
loraine.mendes.ext@cits.br

Kassyane Nunes da Silva
CITS
kassyane.silva@cits.br

Orlando Renato Brenner Lantmann
CITS; UTFPR
orlando.lantmann@cits.br

Hélio Gomes de Carvalho
CITS
helio@utfpr.edu.br

Gustavo Dambiski Gomes de Carvalho
CITS; UNISENAI
gustavo.dambiski@gmail.com

RESUMO

A transformação digital 4.0, que traz oportunidades de inovação, envolve diversas tecnologias habilitadoras, dentre as quais se destaca a Inteligência Artificial (IA). Neste contexto, este trabalho tem como objetivo identificar linguagens de programação, bem como bibliotecas e ferramentas, focadas em aplicações de IA. Quatro linguagens de programação para IA foram identificadas como destaque: C#, R, MatLab e Python. Em seguida, foram abordadas diversas bibliotecas e ferramentas para IA da linguagem Python como Numpy, Pandas, TensorFlow, PyTorch, Scikit-Learn, Keras, bem como frameworks focados em Processamento de Linaguagem Natural (PLN) como LangChain, HuggingFace Transformers, NLTK e SpaCy. Por fim, a conclusão do trabalho consolida os conhecimentos sobre as linguagens de programação, bibliotecas e ferramentas de IA em quadros comparativos com pontos positivos e negativos, os quais organizam a literatura e podem ser utilizados por profissionais e pesquisadores no tema.

Palavras-chave: Inteligência Artificial; IA; Processamento de Linguagem Natural; PLN; Bibliotecas.

1 CONTEXTUALIZAÇÃO

A Transformação Digital 4.0, caracterizada principalmente pela quarta revolução industrial, ou Indústria 4.0 (Schumacher; Erol; Sihn, 2016), possibilita muitas oportunidades de inovação para as organizações (Machado; Schroeder; Carvalho, 2022). Essa revolução digital é caracterizada por um conjunto de tecnologias habilitadoras como Internet das Coisas (Internet of Things – IoT), Impressão 3D, Big Data e Analytics, Computação em Nuvem (Cloud Computing), Robótica Avançada, Inteligência Artificial (Artificial Intelligence - AI), entre outras. Uma definição de Transformação Digital, que inclui aspectos para além da tecnologia, é fornecida por CITS (2023, local. 1):

> "É o processo contínuo das organizações (de comércio, indústria ou serviços) para integrar tecnologias habilitadoras em todas as áreas, por meio de mudanças fundamentais em estratégia, liderança, estrutura, pessoas, processos, cultura e negócios digitais, com o objetivo de suportar as tomadas de decisões a partir de dados, reduzir desperdícios e agregar valor aos seus clientes e à sociedade." (CITS, 2023, local. 1)"

Em especial, nos últimos anos, o avanço da Inteligência Artificial (IA) e do Processamento de Linguagem Natural (PLN) tem revolucionado diversos setores da indústria, pesquisa e desenvolvimento. Com a crescente demanda por soluções inteligentes e automatizadas, a escolha da linguagem de programação e das ferramentas adequadas tornou-se um aspecto crucial para o sucesso de projetos (Ludermir, 2021).

Neste contexto, o Centro Internacional de Tecnologia de Software, que tem um histórico de contribuição ao sistema local de inovação de Curitiba-PR e Manaus-AM (Shimajv; Lorenzi, 2005), constituiu times técnicos especializados para pesquisar e desenvolver os temas de Transformação Digital 4.0 e suas tecnologias habilitadoras, como a Inteligência Artificial (IA). Assim, este trabalho tem como objetivo identificar linguagens de programação, bem como bibliotecas e ferramentas, focadas em aplicações de Inteligência Artificial. Assim, este trabalho contribui para a teoria ao organizar esses conhecimentos, bem como para a prática de profissionais e pesquisadores, que podem consultar este material ao escolher as linguagens de programação, bibliotecas e ferramentas de IA e PLN mais adequadas para seus projetos.

Além desta introdução, a segunda seção descreve brevemente as principais etapas desenvolvidas e a terceira seção apresenta os principais resultados e aprendizados obtidos.

2 DESENVOLVIMENTO

Para atingir o objetivo deste trabalho, primeiramente foram identificadas algumas linguagens de programação utilizadas para Inteligência Artificial como C#, R, MATLAB e Python. Adicionalmente, as principais vantagens e desvantagens de cada linguagem foram apresentadas.

Em seguida, uma vez identificado que a linguagem de programação para IA mais disseminada atualmente é o Python, foram analisadas as principais bibliotecas e ferramentas de Inteligência Artificial (IA) derivadas ou compatíveis com esta linguagem como Numpy, Pandas, TensorFlow, PyTorch, Scikit-Learn, Keras, bem como frameworks focados em Processamento de Linaguagem Natural (PLN) como LangChain, HuggingFace Transformers, NLTK e SpaCy. A Tabela 1 apresenta as principais etapas de desenvolvimento deste trabalho.

Tabela 1: Etapas do desenvolvimento do relato de prática

Etapa	Descrição	Fonte
1	Identificar linguagens de programação para IA e PLN	Literatura
2	Identificar bibliotecas e ferramentas para IA e PLN	Literatura

Fonte: autoria própria (2024).

3 RESULTADOS, DESAFIOS E APRENDIZADO

3.1 LINGUAGENS DE PROGRAMAÇÃO PARA INTELIGÊNCIA ARTIFICIAL

A importância da escolha da linguagem de programação é abordada no trabalho de Duffany (2014), no qual demonstra que a decisão deve considerar a facilidade de aprendizado, o custo, o ambiente de programação e os objetivos do projeto. Tradicionalmente, linguagens compiladas como C++, C e LISP eram as mais populares entre cientistas e desenvolvedores de aplicações científicas e análises de dados. Porém, com o passar dos anos, o uso dessas linguagens compiladas tem diminuído em favor de ambientes interpretados, como MATLAB, R e Python. O aumento desta popularidade é atribuído a várias razões, incluindo a sintaxe simples e limpa, integração eficiente de simulação e visualização, feedback imediato de comandos e a grande disponibilidade e variedade de funções e bibliotecas integradas (Nagpal; Gabriani, 2019).

O avanço da tecnologia tem impulsionado o desenvolvimento de aplicações que se beneficiam em muito da utilização de IA e a escolha da linguagem de programação adequada desempenha um papel fundamental na eficiência e sucesso desses projetos. As linguagens de programação abordadas no presente estudo são: C#, R, MatLab e Python. Essas linguagens são muito populares na ciência e na indústria (Olivari; Olivari, 2022). De acordo como índice TIOBE (2024), que mensura a popularidade de linguagens de programação, as linguagens C#, R, MatLab e Python estão classificadas entre as 20 mais populares em abril de 2024.

Desenvolvida pela Microsoft, C# é uma linguagem orientada a objetos, amplamente utilizada no desenvolvimento de aplicativos corporativos e sistemas de software (Albahari; Albahari, 2017). Com uma sintaxe semelhante ao C++, o C# oferece recursos avançados de tipagem estática e suporte para programação assíncrona, tornando-o adequado para o desenvolvimento de aplicativos escaláveis e robustos (Richter, 2012). Sua integração com o framework .NET proporciona um ambiente de desenvolvimento coeso e rico em recursos para os desenvolvedores (Liberty, 2017).

A linguagem R foi desenvolvida especificamente para análise estatística e visualização de dados. Com uma origem na linguagem S, a linguagem R oferece uma ampla variedade de ferramentas para análise exploratória de dados, modelagem estatística e geração de gráficos (Wickham, 2016). Sua comunidade ativa de usuários e desenvolvedores contribui com novas ferramentas e funcionalidades, expandindo suas capacidades para atender às demandas em constante evolução da ciência de dados (Grolemund; Wickham, 2016).

O MatLab é uma linguagem de programação e ambiente de desenvolvimento amplamente utilizado em engenharia, matemática e ciências aplicadas (Attaway, 2013). Conhecido por sua facilidade de uso e poder computacional, MatLab oferece diversas ferramentas para análise numérica, modelagem e simulação de sistemas dinâmicos (Chapman, 2017). Sua integração com o .NET Framework e suas ferramentas robustas para otimização e modelagem tornam MatLab uma escolha popular entre engenheiros e cientistas em todo o mundo (Palm, 2017).

O Python é uma linguagem de alto nível, conhecida por sua simplicidade e facilidade de aprendizado (Mcfarland, 2024). Amplamente utilizada em ciência de dados e análise estatística, o Python oferece uma diversidade de bibliotecas e frameworks, como NumPy, Pandas e TensorFlow, que aceleram o desenvolvimento de aplicações inteligentes (Smith; Brwon; Johnson, 2017). Sua sintaxe limpa e legível, juntamente com uma comunidade ativa de desenvolvedores, tornam o Python uma escolha popular para projetos em diversas áreas, desde

desenvolvimento web até automação de tarefas (Jones, 2016).

De acordo com o Índice de Comunidade de Programação da Tiobe (Tabela 2), que mede a popularidade das linguagens de programação, o Python ocupa a primeira posição em abril de 2024 (Tiobe, 2024). Desde meados de 2018 a popularidade do Python vem aumentado consideravelmente em contraste a outras linguagens.

Tabela 2: Classificação do Índice TIOBE para abril de 2024.

Ranking (posição)	Linguagem de Programação (unidade)	Ratings (%)
1	Python	16,41
2	C	10,21
3	C++	9,76
4	Java	8,94
5	C#	6,77
6	JavaScript	2,89
7	Go	1,85
8	Visual Basic	1,70
9	SQL	1,61
10	Fortran	1,47
11	Delphi	1,47
12	Assembly	1,30
13	Ruby	1,24
14	Swift	1,23
15	Scratch	1,14
16	MatLab	1,11
17	PHP	1,09
18	Kotlin	1,05
19	Rust	1,03
20	R	0,84

Fonte: Adaptado de Tiobe (2024).

Segundo Nagpal e Gabriani (2019), o Python também se destaca por sua habilidade de interoperar com software legado escrito em C, C++ e outras linguagens, tornando-o uma escolha menos complexa em comparação com outros ambientes de programação. As bibliotecas e frameworks disponíveis atendem praticamente a todas as necessidades em projetos de IA e reduzem significativamente a complexidade da implementação. Além disso, sua natureza orientada a objetos e compatibilidade com outras linguagens o tornam uma escolha ideal para o manuseio eficiente de dados. A comunidade Python, em constante expansão junto com a popularidade da linguagem, garante amplo suporte no desenvolvimento na área de IA.

3.1 BIBLIOTECAS E FERRAMENTAS EM PYTHON PARA IA E PLN

A inteligência artificial engloba uma variedade de técnicas que buscam capacitar computadores a realizar tarefas que, historicamente, exigiriam inteligência humana. Isso inclui o Machine Learning (ML), Processamento de Linguagem Natural (PLN), síntese de linguagem, visão computacional, robótica, dentre outros. O ML, por sua vez, consiste em um conjunto de técnicas que permitem que sistemas computacionais aprendam com dados históricos e melhorem seu desempenho em tarefas específicas. As técnicas de ML abrangem uma variedade de algoritmos, como Support Vector Machines (SVM), árvores de decisão, aprendizado bayesiano, clustering, k-means, aprendizado de regras de associação, regressão e redes neurais. As Redes Neurais (Neural Netowrks - NNs), também conhecidas como Redes Neurais Artificiais, são um subconjunto do ML, inspiradas em redes neurais biológicas. Organizadas em camadas de neurônios artificiais, desempenham um papel fundamental no campo do Deep Learning (DL). As arquiteturas típicas de DL incluem redes neurais profundas (DNNs), redes neurais convolucionais (CNNs), redes neurais recorrentes (RNNs) e redes generativas adversárias (GANs) (Nguyen et al., 2019).

Em resposta à crescente diversidade de técnicas, surgiram bibliotecas e frameworks avançados de IA. Esses recursos têm como objetivo auxiliar os desenvolvedores na tarefa de aplicar códigos complexos. As bibliotecas oferecem módulos e funcionalidades específicas, enquanto os frameworks fornecem estruturas abrangentes para o desenvolvimento eficiente de soluções de IA. A evolução destas ferramentas facilita a implementação de algoritmos, acelerando o processo de criação e aprimoramento de sistemas inteligentes. A seguir são apresentadas algumas destas bibliotecas e ferramentas disponíveis para linguagem Pyhton como Pandas, Numpy, SciPy, TensorFlow, PyTorch, Scikit-Learn, Keras, LangChain, HuggingFace, NLTK e spaCy (Grus, 2016; Mcfarland, 2024; Melnik, 2024).

Pandas: é uma biblioteca fundamental no Python, especialmente no contexto do ML. Sua função principal é atuar como uma ferramenta de análise de dados, oferecendo recursos robustos para manipulação e exploração de conjuntos de dados multidimensionais estruturados, incluindo séries temporais. Por meio de estruturas de dados como Series e DataFrames, o Pandas permite aos desenvolvedores trabalharem de forma eficiente com dados, oferecendo funcionalidades como indexação, alinhamento, fusão e junção de conjuntos de dados, além de diversas operações para manipulação e análise dos dados (Grus, 2016; Mcfarland, 2024; Melnik, 2024).

Numpy: é outra biblioteca muito utilizada para ML e IA. Reconhecido por sua eficiência e desempenho, o NumPy é amplamente utilizado para realizar operações matemáticas em matrizes multidimensionais. Sua estrutura de dados, o objeto de matriz N-dimensional, oferece alta performance e eficácia na manipulação e análise de dados. Além disso, o NumPy proporciona recursos para manipular a forma das matrizes, realizar limpeza e manipulação de dados, e executar operações estatísticas e de álgebra linear. Possui capacidade de otimizar o desempenho dos modelos de ML, sem exigir uma complexidade excessiva (Grus, 2016; Mcfarland, 2024; Melnik, 2024).

SciPy: é uma biblioteca gratuita de código aberto, que se baseia no NumPy e oferece funcionalidades avançadas para computação científica e técnica. Projetado especialmente para lidar com grandes conjuntos de dados, possui módulos embutidos para otimização de array e álgebra linear, semelhantes aos do NumPy. Além disso, enriquece as funções do NumPy, transformando-as em ferramentas científicas de fácil uso. Comumente empregado na manipulação de imagens, fornece recursos essenciais para uma variedade de funções matemáticas de alto nível (Mcfarland, 2024; Melnik, 2024).

TensorFlow: trata-se de uma biblioteca de DL de código aberto desenvolvida pelo Google, sendo amplamente reconhecido como uma das principais estruturas para computação numérica usando grafos de fluxo de dados. Originalmente criado pelos pesquisadores do Google Brain, o TensorFlow foi disponibilizado ao público como uma biblioteca de código aberto. Uma de suas principais vantagens é a capacidade de computação distribuída, especialmente entre várias GPUs, tornando-o uma escolha popular para treinamento de modelos em larga escala. Uma adição significativa ao TensorFlow foi a incorporação do Keras em sua segunda versão, proporcionando uma API mais amigável para desenvolvedores. Embora historicamente tenha sido considerado difícil de aprender e exigisse muito código para criar modelos, várias melhorias foram feitas ao longo dos anos para abordar essas preocupações. A disponibilidade de mais materiais de aprendizado, tutoriais aprimorados e cursos online têm tornado o TensorFlow mais acessível aos desenvolvedores. Além disso, novas seções da biblioteca, como tf.keras e tf.estimator, oferecem interfaces de alto nível para facilitar a criação e treinamento de modelos. O TensorFlow Lite é uma versão leve destinada a dispositivos móveis e incorporados, permitindo a inferência de aprendizado de máquina com baixa latência e tamanho binário reduzido. Ele também suporta aceleração de hardware com a API Android Neural Networks. Com uma arquitetura flexível, suporte multiplataforma e capacidades de abstração, o TensorFlow gerencia DNNs de forma eficaz, tornando-se uma ferramenta essencial

para projetos de IA e ML (Mcfarland, 2024; Melnik, 2024).

PyTorch: o PyTorch é uma biblioteca de ML de código aberto conhecida por seu gráfico computacional dinâmico, distinguindo-se por seus tensores e redes neurais dinâmicas em Python com forte aceleração de GPU. Os tensores, uma construção matemática fundamental, são amplamente usados em Física e Engenharia, representando matrizes especiais que podem gerar novos vetores com diferentes magnitudes e direções. A aceleração de GPU, essencial para as DNNs modernas, é plenamente suportada pelo PyTorch. Além disso, permite a criação de redes neurais dinâmicas, que podem ser ajustadas a cada iteração do treinamento, possibilitando a adição ou remoção de camadas ocultas para melhorar a precisão e a generalização do modelo. Enquanto o TensorFlow adota uma abordagem de construção de grafo de fluxo de dados único, otimizando o código do grafo para desempenho, o PyTorch recria o grafo em tempo real a cada iteração, oferecendo uma maior flexibilidade. Ele integra bibliotecas de aceleração como Intel MKL, Nvidia cuDNN e NCCL para maximizar a velocidade, utilizando seus principais backends como bibliotecas independentes com uma API (Mcfarland, 2024; Melnik, 2024).

Scikit-Learn: também conhecido como sklearn, é uma biblioteca de ML de código aberto amplamente utilizada na comunidade Python. Desenvolvida com uma interface simples e eficiente, o Scikit-Learn oferece uma ampla gama de algoritmos de ML para tarefas de classificação, regressão, agrupamento e pré-processamento de dados. Sua popularidade se deve em grande parte à sua facilidade de uso e à sua integração perfeita com outras bibliotecas Python, como NumPy, Pandas e Matplotlib (Grus, 2016; Mcfarland, 2024; Melnik, 2024).

Keras: o Keras é uma biblioteca de alto nível para o desenvolvimento e avaliação de NN em modelos de ML e DL. Funciona como uma API de redes neurais que roda em cima de frameworks como TensorFlow. Permite que os desenvolvedores experimentem rapidamente e convertam ideias em resultados de forma eficiente. Ele oferece suporte para a construção de redes neurais modulares, com ênfase especial em CNNs, o que é valioso para aplicações de visão computacional. Além disso, o Keras facilita a construção de arquiteturas de rede mais complexas, como GoogLeNet e SqueezeNet, através da possibilidade de construir redes baseadas em sequência ou em grafos. A biblioteca é de código aberto e oferece suporte para execução em várias plataformas. Suas principais características incluem agrupamento de dados, desenvolvimento de camadas neurais, construção de modelos de DL e ML, além de oferecer uma ampla gama de funções de ativação e custo (Mcfarland, 2024; Melnik, 2024).

LangChain: é um framework de código aberto projetado para simplificar o desenvolvimento de aplicativos impulsionados por modelos de linguagem, com foco especial

nos LLMs (Large Language Models). Diferenciando-se das chamadas de API convencionais, o LangChain é capaz de realizar operações sofisticadas e age como um agente inteligente, permitindo conexões flexíveis com diversas fontes de dados. Essa abordagem enriquece as experiências do usuário, tornando-as mais personalizadas e contextuais. O LangChain acelera o desenvolvimento de aplicações como chatbots, sistemas de perguntas e respostas generativas (GQA) e sumarização automática (Mcfarland, 2024; Melnik, 2024).

HuggingFace Transformers: o HuggingFace Transformers é um framework de código aberto, desenvolvido e mantido pela comunidade e pela HuggingFace, com o propósito de fornecer acesso simplificado a modelos de PLN de última geração. Com suporte para PyTorch, TensorFlow e JAX, essa plataforma oferece uma ampla variedade de modelos pré-treinados para diversas tarefas em modalidades como texto, visão e áudio. Sua versatilidade e capacidade de disponibilizar milhares de modelos pré-treinados o tornam uma ferramenta indispensável para pesquisadores, desenvolvedores e cientistas de dados que buscam explorar e aplicar os mais recentes avanços em PLN. O HuggingFace Transformers não apenas simplifica o acesso a esses modelos, mas também promove a colaboração e o desenvolvimento comunitário, impulsionando assim a inovação e o progresso contínuo na área de PLN (Mcfarland, 2024; Melnik, 2024).

NLTK: é uma biblioteca amplamente utilizada para PLN em Python. Desenvolvido pela Universidade de Tecnologia de Sidney, o NLTK oferece uma ampla gama de ferramentas e recursos para a análise e manipulação de texto em linguagem humana. Desde sua introdução, o NLTK se tornou uma ferramenta essencial para pesquisadores, desenvolvedores e profissionais de PLN em todo o mundo, devido à sua facilidade de uso, extensibilidade e eficácia em várias tarefas de PLN, como tokenização, stemming, lematização, análise sintática, análise de sentimento, etc. Sua vasta coleção de modelos pré-treinados o torna uma escolha popular para testes e desenvolvimento de modelos de ML voltados para o processamento de texto. Em resumo, o NLTK desempenha um papel importante no avanço da pesquisa e aplicação prática de técnicas de PLN, impulsionando inovações em áreas como tradução automática, sumarização de texto, entre outros (Mcfarland, 2024; Melnik, 2024).

spaCy: é uma biblioteca de PLN de código aberto, altamente otimizada e eficiente, desenvolvida para oferecer um conjunto abrangente de ferramentas para tarefas de PLN. Com sua arquitetura modular e fácil de usar, o spaCy permite o processamento rápido e preciso de texto em larga escala. Uma de suas principais características é a capacidade de realizar tokenização, análise morfológica, reconhecimento de entidades nomeadas (NER), análise de

ISSN: 2594-8083

dependência sintática e muito mais, tudo em um único pacote. Além disso, o spaCy é fornecido com modelos pré-treinados para vários idiomas, o que facilita ainda mais o desenvolvimento de aplicativos de PLN em diferentes contextos linguísticos (Mcfarland, 2024; Melnik, 2024).

Assim, essas bibliotecas Python desempenham um papel fundamental no desenvolvimento de aplicações de IA, oferecendo recursos que aceleram significativamente o processo de construção e implantação. A sintaxe simples e legível do Python, juntamente com sua popularidade e suporte ativo da comunidade, torna-o uma escolha preferencial para profissionais e pesquisadores que buscam criar soluções inovadoras com IA (Mcfarland, 2024; Melnik, 2024).

3.3 CONCLUSÃO E APRENDIZADO

O objetivo deste trabalho foi atingido, uma vez que este buscou identificar linguagens de programação, bem como bibliotecas e ferramentas, focadas em aplicações de Inteligência Artificial. Nesse sentido, para consolidar o aprendizado relatado neste trabalho, a Tabela 3 compara pontos positivos e negativos das linguagens de programação para IA identificadas: C#, R, MatLab e Python. A evolução das linguagens de programação e o cenário em constante mudança das tecnologias de desenvolvimento refletem o crescimento de novas linguagens de programação, especialmente no âmbito de aplicação para a Inteligência Artificial. A ascensão do Python ao topo do Índice de Comunidade de Programação da TIOBE, corroborada pela sua simplicidade, versatilidade e extensa variedade de bibliotecas, destaca seu papel fundamental no desenvolvimento de projetos científicos e analíticos. A interoperabilidade do Python com outras linguagens e sua capacidade de lidar eficientemente com dados contribuem para sua crescente utilização em projetos de IA e ML.

Por sua vez, a Tabela 4 demonstra resumidamente as características das principais ferramentas disponíveis para se utilizar no Python para o desenvolvimento de IA e ML. A disponibilidade de bibliotecas e frameworks avançados, acompanhado do surgimento de novas ferramentas demonstram o compromisso contínuo da comunidade em simplificar e fortalecer o desenvolvimento de soluções de IA. Essas inovações prometem acelerar ainda mais o progresso da IA, capacitando pesquisadores e desenvolvedores a explorar os mais recentes avanços e aplicá-los de forma eficaz em uma variedade de projetos

Tabela 3: Comparação entre linguagens de programação para IA

Ling.	Pontos Positivos	Pontos Negativos
C#	Orientada a objetos, adequada para o desenvolvimento de aplicativos corporativos.	Requer o ambiente .NET Framework para execução, o que pode adicionar dependências e aumentar o tamanho da distribuição do aplicativo.
	Recursos avançados de tipagem estática e suporte para programação assíncrona.	Limitações em plataformas não-Windows, o que pode restringir a portabilidade do código em determinados casos.
	Integração com o framework .NET proporcionando um ambiente de desenvolvimento coeso e rico em recursos.	Demanda mais tempo para aprendizado em comparação com linguagens mais simples, como Python ou JavaScript.
R	Especializada em análise estatística e visualização de dados.	Aprendizado mais demorado para iniciantes devido à sintaxe específica e peculiaridades da linguagem.
	Ampla variedade de ferramentas para análise exploratória de dados e modelagem estatística.	Menos adequada para desenvolvimento de aplicativos além do escopo de análise de dados.
	Comunidade ativa contribuindo com novas ferramentas e funcionalidades.	Desempenho inferior em comparação com linguagens compiladas como C++ ou Java.
MATLAB	Facilidade de uso e poder computacional para análise numérica, modelagem e simulação de sistemas dinâmicos.	Licenciamento comercial pode ser caro, especialmente para uso em larga escala em organizações ou projetos acadêmicos.
	Integração com o .NET Framework e ferramentas robustas para otimização e modelagem.	Menos adequada para desenvolvimento de aplicativos complexos ou de grande escala, em comparação com linguagens de propósito geral como Python ou Java.
	Popular entre engenheiros e cientistas em engenharia, matemática e ciências aplicadas.	Necessário um período inicial de adaptação a linguagem mais longo quando comparado às outras linguagens.
Python	Simplicidade e facilidade do aprendizado.	Desempenho relativamente mais lento em comparação com linguagens de baixo nível.
	Diversidade de bibliotecas e frameworks especializados em ciência de dados e IA.	Gerenciamento de memória automático pode resultar em *overheads* e consumo de recursos adicionais.
	Sintaxe limpa e legível. Comunidade ativa de desenvolvedores.	Limitações em aplicações que requerem alto desempenho, como processamento de grandes volumes de dados em tempo real.

Fonte: autoria própria.

Tabela 4: Comparação entre bibliotecas e ferramentas de IA e ML

Ferramenta	Pontos Positivos	Pontos Negativos
Pandas	- Facilita a manipulação e análise de dados estruturados	- Pode ter desempenho inferior em comparação com outras ferramentas para grandes conjuntos de dados
NumPy	- Eficiente para operações matemáticas em matrizes multidimensionais	- Curva de aprendizado pode ser extensa para iniciantes
SciPy	- Oferece funcionalidades avançadas para computação científica e técnica	- Documentação menos amigável para iniciantes
TensorFlow	- Suporte para computação numérica usando grafos de fluxo de dados	- Curva de aprendizado íngreme para iniciantes
PyTorch	- Gráfico computacional dinâmico e suporte para tensores e redes neurais dinâmicas em Python	- Menos otimizado para computação distribuída em comparação com o TensorFlow
Scikit-Learn	- Interface simples e eficiente para uma ampla gama de algoritmos de aprendizado de máquina	- Pode não ser adequado para tarefas extremamente específicas que exigem algoritmos personalizados
Keras	- Facilita o desenvolvimento e avaliação de redes neurais em modelos de ML e DL	- Pode ser menos flexível em comparação com outras bibliotecas para a criação de arquiteturas de rede altamente personalizadas
LangChain	- Simplifica o desenvolvimento de aplicativos impulsionados por modelos de linguagem, com conexões flexíveis com diversas fontes de dados	- Pode ter uma curva de aprendizado mais íngreme devido à sua abordagem única
HuggingFace	- Fornece acesso simplificado a modelos de PLNl de última geração	- Pode exigir conhecimento prévio de *frameworks* como PyTorch ou TensorFlow para aproveitar totalmente suas capacidades
NLTK	- Ampla gama de ferramentas e recursos para análise e manipulação de texto em linguagem humana	- Pode ser menos eficiente em comparação com bibliotecas mais recentes para tarefas específicas de PLN
spaCy	- Arquitetura eficiente e modular para processamento rápido e preciso de texto em larga escala	- Alguns recursos podem exigir treinamento adicional para atingir a máxima precisão

Fonte: autoria própria.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio do Centro Internacional de Tecnologia de Software - CITS.

REFERÊNCIAS

ATTAWAY, S. **MATLAB: A Practical Introduction to Programming and Problem Solving**. Elsevier, 2013.

ALBAHARI, J.; ALBAHARI, B. **C# 7.0 in a Nutshell: The Definitive Reference**. O'Reilly Media, 2017.

CHAPMAN, S. **MATLAB Programming for Biomedical Engineers and Scientists**. Academic Press, 2017

CITS – CENTRO INTERNACIONAL DE TECNOLOGIA DE SOFTWARE. **Pesquisa e Desenvolvimento**. Disponível em: <cits.br>. Acesso em: 02 dez. 2023

DUFFANY, J. L. Choice of Language for an Introduction to Programming Course. **Latin American and Caribbean Conference for Engineering and Technology.** pp. 1-9, 2014.

GROLEMUND, G.; Wickham, H. **R for Data Science**. O'Reilly Media, 2016.

GRUS, Joel. **Data science do zero**. Alta books, 2016.

JONES, E. **Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython**. O'Reilly Media, 2016.

LIBERTY, J. **Learning C# by Developing Games with Unity 5.x**. Packt Publishing, 2017.

MACHADO, A. C. M.; SCHROEDER, R. T.; CARVALHO, G. D. G. Transformação Digital 4.0 em cooperativas. In: Carvalho, H. G., Diogo, T. M., et al (Ed.). **Gestão da Inovação em cooperativas: um caminho para inovar**. Curitiba: ISAE, 2022, p.266-283.

MCFARLAND, A. **10 melhores bibliotecas Python para aprendizado de máquina e IA12**. Unite.AI, 2024. Disponível em < https://www.unite.ai/pt/10-melhores-bibliotecas-python-para-aprendizado-de-m%C3%A1quina-ai/>. Acesso em: 21 mar. 2024.

MELNIK, Y. The **Top 16 AI Frameworks and Libraries: A Beginner's Guide**. DataCamp Blog, 2023. Disponível em: < https://www.datacamp.com/blog/top-ai-frameworks-and-libraries>. Acesso em: 21 mar. 2024.

NAGPAL, Abhinav; GABRANI, Goldie. Python for data analytics, scientific and technical applications. **Amity international conference on artificial intelligence (AICAI)**. IEEE,

2019. p. 140-145, 2019.

NGUYEN, Giang et al. Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. **Artificial Intelligence Review**, v. 52, p. 77-124, 2019.

OLIVARI, Luka; OLIVARI, Luca. Influence of Programming Language on the Execution Time of Ant Colony Optimization Algorithm. **Tehnički glasnik**, v. 16, n. 2, p. 231-239, 2022.

PALM, W. J. **MATLAB for Engineers**. McGraw-Hill Education, 2017.

RICHTER, J. **CLR via C#**. Microsoft Press, 2012.

SCHUMACHER, Andreas; EROL, Selim; SIHN, Wilfried. A maturity model for assessing Industry 4.0 readiness and maturity of manufacturing enterprises. **Procedia Cirp**, v. 52, p. 161-166, 2016.

SHIMAJV, Walter Tadahiro; LORENZI, Antonio Guilherme de Arruda. O papel do CITS1 na política de desenvolvimento tecnológico no Paraná. **Production**, v. 15, p. 310-321, 2005.

SMITH, J.; BROWN, K.; JOHNSON, L. **Python for Data Science for Dummies**. Wiley, 2017.

TIOBE. **Tiobe index**. 2024. Disponível em: < https://www.tiobe.com/tiobe-index/>. Acesso em: mar. 2024.

WICKHAM, H. **ggplot2: Elegant Graphics for Data Analysis**. Springer, 2016.