Investigando o Uso da Inteligência Artificial em Projetos Python Hospedados no GitHub

Luiz Andre do Nascimento Ubaldo[1], **Jailton Coelho**[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Paraná (IFPR)
Campus Telêmaco Borba – Telêmaco Borba, PR – Brasil

202130177555@estudantes.ifpr.edu.br, jailton.coelho@ifpr.edu.br

Abstract. *Artificial Intelligence (AI) has evolved significantly in recent years. Despite the growing popularization of AI, has it also been incorporated into the development of open-source projects in recent years? Motivated by this question, a study with 15,770 Python repositories was conducted. The results showed that the most used Python libraries for AI were TensorFlow, OpenCV, and Scikit-Learn. It was also observed that 12% of the projects have at least one dependency on an AI-related library. Finally, it was observed that the countries with the highest number of Python projects related to AI are China, the United States, and Germany.*

Resumo. *A Inteligência Artificial (IA) tem evoluído significativamente nos últimos anos. Apesar da crescente popularização da IA, será que ela também tem sido incorporada ao desenvolvimento de projetos de código-aberto nos últimos anos? Sob esta motivação, foi realizado um estudo com 15.770 repositórios Python. Os resultados mostraram que as bibliotecas em Python para a área de IA mais usadas foram TensorFlow, OpenCV e Scikit-Learn. Observou-se também que 12% dos projetos possuem pelo menos uma dependência para uma biblioteca relacionado à IA. Por fim, observou-se que os países com o maior número de projetos Python relacionados à IA são China, Estados Unidos e Alemanha.*

1. Introdução

A inteligência artificial (IA) está avançando rapidamente, transformando diversos aspectos da vida moderna. A partir de 2012, a área experimentou um novo ciclo de expansão impulsionado pelo aprendizado profundo. Esta fase, conhecida como a ”nova revolução da IA”, tem aberto portas para aplicações inovadoras e transformadoras em vários setores [Tang 2018].

Recentemente, o interesse em IA tem crescido exponencialmente, em parte devido ao lançamento público do ChatGPT [Dakhel et al. 2023]. Ferramentas de geração de código assistido por IA têm se tornado cada vez mais comuns na computação, oferecendo a capacidade de gerar código a partir de comandos em linguagem natural ou entradas de código parciais. Exemplos significativos dessas ferramentas incluem o GitHub Copilot, o Amazon CodeWhisperer e o ChatGPT da OpenAI. Essas ferramentas ajudam a reduzir a necessidade de escrita manual de código e a tornar o processo de desenvolvimento mais eficiente e menos demorado. Existem diversas abordagens para a geração automática de código, como a programação

em linguagem natural [Shin and Nam 2021, Wong et al. 2023], modelos formais [Gomes and Baunach 2019], Algoritmos Evolutivos [Slowik and Kwasnicka 2020] e Modelos de Linguagem de Grande Escala (LLMs) [Thirunavukarasu et al. 2023, Fan et al. 2023, Chang et al. 2023, Zhao et al. 2023].

Apesar da popularização da inteligência artificial nos últimos anos, será que ela também passou a integrar o desenvolvimento de projetos de código-aberto? É sob esta motivação que este estudo busca responder às seguintes questões de pesquisa:

QP1. Quais são as bibliotecas e frameworks mais utilizados em Python na área de inteligência artificial?

QP2. Qual é o percentual de projetos desenvolvidos em Python que dependem de alguma biblioteca ou framework relacionado à inteligência artificial?

QP3. Quais são as características dos projetos Python que utilizam inteligência artificial em seu desenvolvimento?

QP4. Quais países desenvolvem mais projetos em Python relacionados à inteligência artificial?

Para responder a essas questões de pesquisa, foram analisados 15.770 repositórios Python hospedados no GitHub, selecionados entre os 100 mil repositórios mais populares.

O restante deste artigo está organizado conforme descrito a seguir. A Seção 2 descreve como foi realizada a coleta das informações. A Seção 3 apresenta os resultados obtidos pela análise dos dados. A Seção 4 discute os trabalhos relacionados. A Seção 5 aborda as ameaças à validade deste estudo. Por fim, a Seção 6 apresenta as conclusões finais.

2. Procedimentos Metodológicos

Primeiramente, foi criada uma lista com os 100.000 repositórios mais populares do GitHub, ordenados pelo número de estrelas (em abril de 2024). Estrelas são um *proxy* comum para ordenar a popularidade dos projetos do GitHub [Borges et al. 2016, Coelho et al. 2018, Coelho et al. 2020, Coelho 2023]. O estudo limitou-se aos 100.000 repositórios mais populares para focar em projetos mais relevantes. Em seguida, foram filtrados apenas os repositórios cuja linguagem de programação principal fosse Python. Após essa etapa, foram selecionados apenas os repositórios Python que continham o arquivo *requirements.txt* e dependiam de pelo menos uma biblioteca relacionada à inteligência artificial, como TENSORFLOW, PYTORCH, SCIKIT-LEARN, NLTK, entre outras. Todas as bibliotecas e frameworks considerados são apresentados na Tabela 1. A análise de dependências relacionadas a LLMs nos arquivos *requirements.txt* não foram realizadas nesse estudo, pois as LLMs podem ser gerenciados por ferramentas e métodos que não se refletem claramente nesses arquivos, como ambientes virtuais específicos ou contêineres.

No próximo passo, todos os arquivos com extensão *.py* de cada repositório foram inspecionados em busca de algum *import* da biblioteca na qual eles dependiam. O Listing 1 mostra um fragmento de código-fonte em Python de *imports* que poderiam ser encontrados em arquivos com esse tipo de extensão. Neste exemplo, observa-se uma dependência das bibliotecas SCIKIT-LEARN e XGBOOST. Esta etapa foi realizada para aumentar a confiabilidade de que apenas os repositórios que realmente usavam a de-

Table 1. Lista de bibliotecas relacionadas à IA, disponíveis no PyPI, categorizadas por área de aplicação.

Categoria de IA	Biblioteca de IA
Aprendizado de Máquina (ML)	Scikit-Learn TensorFlow CatBoost LightGBM XGBoost MLlib (Spark) Weka LIBSVM LIBLINEAR
Processamento de Linguagem Natural (NLP)	NLTK Gensim SpaCy Hugging-Py-Face AllenNLP Stanford CoreNLP
Aprendizado Profundo (Deep Learning)	PyTorch Keras Theano Caffe MXNet Chainer Fastai
Visão Computacional (CV)	OpenCV Turi Create

pendência encontrada no arquivo *requirements.txt* fossem selecionados. Posteriormente, com 95% de confiança e uma margem de erro de 10%, 88 repositórios foram inspecionados manualmente para garantir a confiabilidade do algoritmo. Todos esses repositórios escolhidos aleatoriamente de fato faziam uso da dependência. Ao final deste processo, foi obtida uma lista de 999 repositórios implementados em Python que apresentavam alguma dependência de bibliotecas ou frameworks relacionados à inteligência artificial.

```
from lightning.app import LightningWork, LightningApp
from sklearn import datasets
from sklearn.model_selection import train_test_split
from xgboost import XGBClassifier
```

Code Listing 1. Exemplo de importação em um arquivo Python.

3. Resultados

QP1. Quais são as bibliotecas e frameworks mais utilizados em Python na área de inteligência artificial?

Para responder a essa primeira questão de pesquisa, foi criada uma lista com a

bibliotecas/frameworks Python relacionadas a inteligência artificial dentre os 100.000 repositórios mais populares do GitHub. Para este estudo foram consideradas as seguintes áreas da inteligência artificial: Machine Learning (ML), Deep Learning (DL), Computer Vision (CV) e Natural Language Processing (NLP). Foram utilizadas apenas os as bibliotecas que tinham alguma dessas classificações no README ou como keyword do seu respectivo repositório e que fizessem parte do pip[1]. A Tabela 1 mostra em qual área da inteligência artificial cada uma das bibliotecas consideradas neste estudo estão relacionadas. Para fazer a contagem das bibliotecas mais usadas, o arquivo *requirements.txt* de cada um dos repositórios foram inspecionados. Um exemplo de arquivo *requirements.txt* pode ser visto no Listing 2.

```
# IA Dependencies
lightning==2.0.0
scikit-learn==1.0.2
xgboost==1.5.2
# Other Dependencies
numpy==1.21.0
pandas==1.3.3
flask==2.0.2
requests==2.26.0
pytest==6.2.5
matplotlib==3.4.3
seaborn==0.11.2
```

Code Listing 2. Exemplo de arquivo requirements.txt

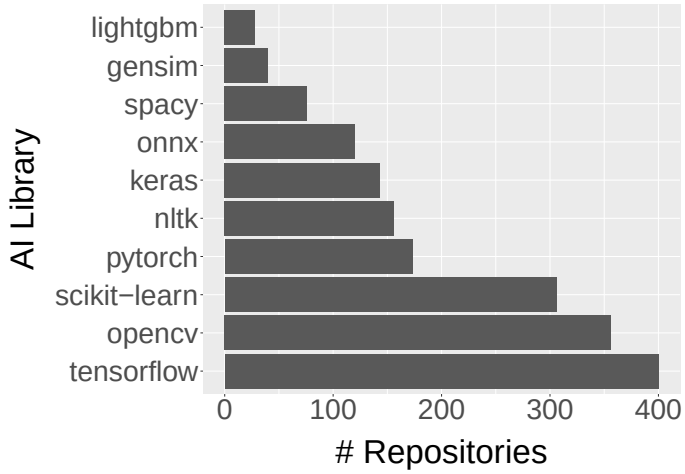

Figure 1. As 10 bibliotecas Python mais frequentemente mencionadas como dependências nos arquivos *requirements.txt* dos repositórios.

A Figura 1 mostra as 10 bibliotecas com o maior número de com o maior número de referências. As bibliotecas que mais se destacaram foram TensorFlow, PyTorch e Scikit-learn com 400, 356 e 306 repositórios dependentes, respectivamente.

1. TensorFlow é uma plataforma de aprendizado de máquina de código aberto para desenvolver modelos a partir de dados de treinamento. TensorFlow oferece uma

[1]https://pypi.org/

extensa coleção de funções e classes que permitem aos usuários construir modelos complexos do zero. TensorFlow possui ampla aceitação na comunidade de aprendizado de máquina devido à sua facilidade de uso, interface Python e capacidade de implantar modelos em navegadores da web e dispositivos móveis. Embora seja principalmente utilizado para aprendizado de máquina, TensorFlow também pode ser empregado no desenvolvimento de tarefas não relacionadas a ML que requerem cálculos numéricos usando grafos de fluxo de dados. No GitHub, o repositório do TensorFlow possui mais de 184.000 estrelas, com mais de 3.500 contribuidores e mais de 200 releases.

2. Scikit-learn é uma biblioteca de aprendizado de máquina gratuita e de código aberto para Python, lançada em 2011. A biblioteca implementa algoritmos de aprendizado supervisionado e não supervisionado bem conhecidos, como regressões linear e logística, máquinas de vetores de suporte, árvores de decisão e clustering k-means. Além disso, a biblioteca oferece técnicas para gerenciar, avaliar e implantar os modelos mencionados anteriormente. No GitHub, o repositório do SCIKIT-LEARN possui mais de 58 mil estrelas, quase 3 mil contribuidores e foi bifurcado mais de 25 mil vezes.

3. OpenCV é uma biblioteca de software de visão computacional e aprendizado de máquina de código aberto. Lançada inicialmente em 1999, o OpenCV evoluiu para se tornar um pilar em diversas aplicações de visão computacional, processamento de imagens e aprendizado de máquina. A biblioteca oferece um conjunto abrangente de funcionalidades que vão desde tarefas básicas de processamento de imagem como filtragem, transformações e extração de características até algoritmos avançados de visão computacional como detecção de objetos, reconhecimento facial e reconhecimento de gestos. O OpenCV, no GitHub, conta com mais de 76 mil estrelas, quase 1.600 contribuidores e mais de 55 mil bifurcações.

QP2. Qual é o percentual de projetos desenvolvidos em Python que dependem de alguma biblioteca ou framework relacionado à inteligência artificial?

Para responder a essa questão de pesquisa foram considerados todos os repositórios Python dentre o 100.000 mais populares. Ao todo, foram analisados 15.770 repositórios através da análise do arquivo *requirements.txt*. A Figura 2 mostra a percentual de repositórios que possuem pelo menos uma dependência (12%) e dos que não possuem (88%) dependência de alguma biblioteca relacionada a inteligencia artificial listada na Tabela 1.

QP3. Quais são as características dos projetos Python que utilizam inteligência artificial em seu desenvolvimento?

A Figura 3 apresenta a distribuição do número de estrelas (*stars*), commits (*commits*) e contribuidores (*contributors*) dos repositórios, com remoção dos *outliers* pelo método Intervalo Interquartil (IQR). Este método identifica como *outliers* os valores que estão significativamente abaixo ou acima da maioria dos dados, comparando-os com a faixa central onde se concentra a maior parte dos valores. Estrelas (*stars*) são exibidas como uma medida aproximada do interesse em um repositório GitHub. *Commits* (*commits*) são alterações registradas no repositório GitHub que compartilham o

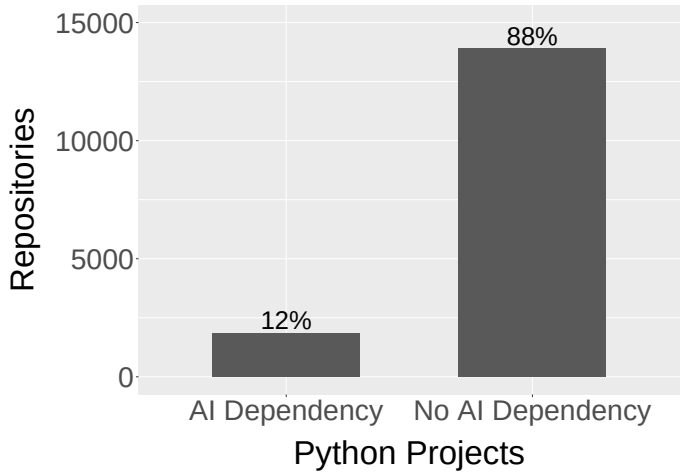

Figure 2. Percentual de projetos Python do GitHub que possuem pelo menos uma dependência de alguma biblioteca relacionada à IA.

mesmo código e configurações de visibilidade com o repositório original. Por fim, os contribuidores (*contributors*) em um repositório GitHub são uma medida do número de usuários que contribuíram com código para o projeto. Os repositórios Python apresentam na mediana 832 estrelas, 80 commits e 4 contribuidores. Os três repositórios mais populares são THEALGORITHMS/PYTHON, AUTOMATIC1111/STABLE-DIFFUSION-WEBUI e KERAS-TEAM/KERAS e possuem 179.418, 129.341 e 60927 estrelas, respectivamente.

O projeto THEALGORITHMS/PYTHON é um repositório no GitHub que contém implementações de diversos algoritmos escritos em Python. Ele serve como um recurso educacional para estudantes e desenvolvedores que desejam aprender mais sobre algoritmos e estruturas de dados. AUTOMATIC1111/STABLE-DIFFUSION-WEBUI é um projeto que oferece uma interface web para o modelo de geração de imagens Stable Diffusion. Ele facilita a interação com o modelo, permitindo aos usuários gerar imagens a partir de texto de forma intuitiva. Por fim, o KERAS-TEAM/KERAS é uma biblioteca de código aberto em Python que fornece uma API para a construção e treinamento de modelos de aprendizado profundo e também pode ser adotada para prototipagem rápida e pesquisa em deep learning.

QP4. Quais países desenvolvem mais projetos em Python relacionados à inteligência artificial?

Para responder a essa quarta e última questão de pesquisa, foi utilizada a lista de 999 repositórios Python que apresentavam pelo menos uma dependência de bibliotecas relacionadas à inteligência artificial (Tabela 1). Entretanto, nem todos os repositórios tinham a localidade preenchida no GitHub; apenas 618 deles foram considerados nesta questão de pesquisa. O gráfico na Figura 4 destaca a localidade desses repositórios, onde os países com mais projetos desenvolvidos são representados em tons mais vivos de vermelho. O gráfico da Figura 4 mostra que os países com mais projetos desenvolvidos são China (169), Estados Unidos (54) e Alemanha (54).

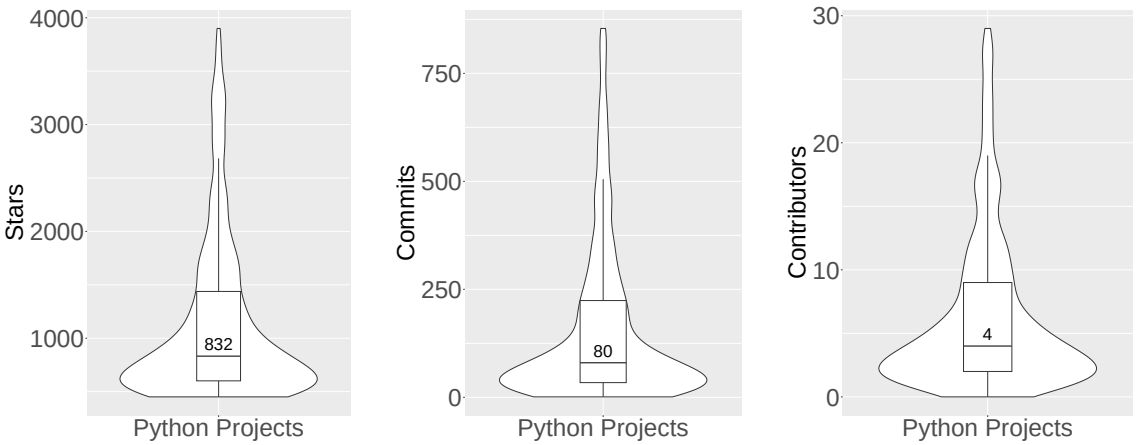

Figure 3. Distribuição do número de estrelas (*stars*), commits (*commits*) e contribuidores (*contributors*) dos repositórios Python no GitHub, com remoção de *outliers*.

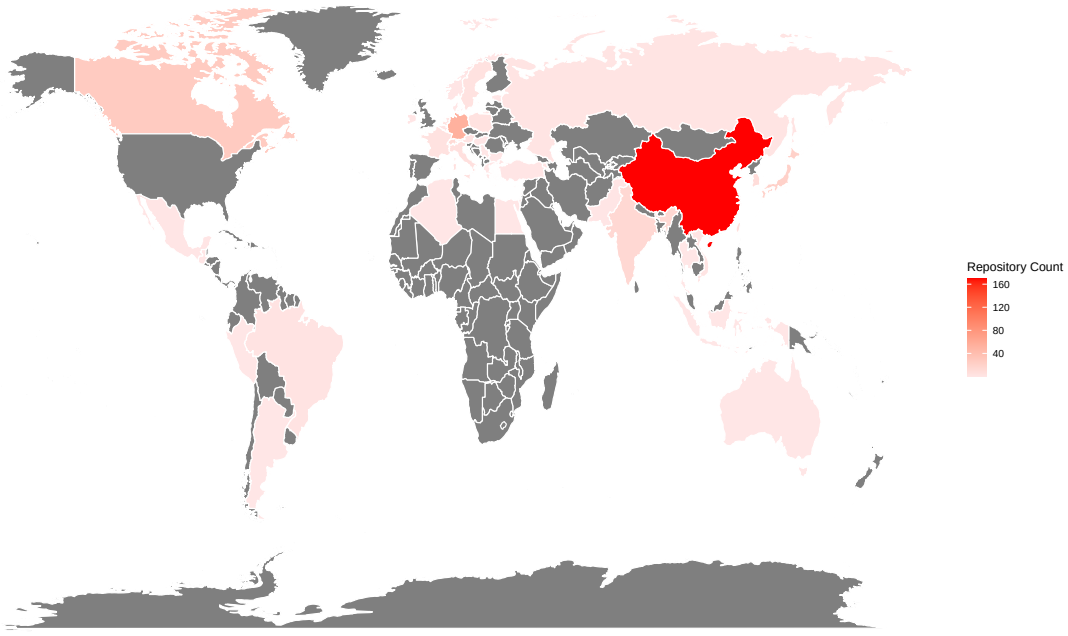

Figure 4. Distribuição geográfica dos repositórios Python relacionados à inteligência artificial, onde os países com mais projetos aparecem em vermelho mais intenso.

4. Trabalhos Relacionados

[Gonzalez et al. 2020] analisaram 700 repositórios de ferramentas de IA e ML e 4.524 repositórios de aplicações no GitHub, aproveitando a rastreabilidade dos dados entre issues, commits, pull requests e usuários. Para comparação, foram incluídos 4.101 repositórios não relacionados. O estudo também examinou o fluxo de trabalho dos desenvolvedores, medindo a colaboração e a autonomia nos repositórios, revelando insights sobre os 10 anos de história dessa comunidade, como a predominância do Python e os repositórios mais populares, como TensorFlow e Tesseract. Os resultados indicam que a

comunidade de IA e ML tem características únicas que devem ser consideradas.

[Peng et al. 2023] realizaram um experimento controlado com o GitHub Copilot, um programador de pares com IA. Desenvolvedores de software recrutados foram solicitados a implementar um servidor HTTP em JavaScript o mais rápido possível. O grupo de tratamento, com acesso ao programador de pares com IA, concluiu a tarefa 55,8% mais rápido do que o grupo de controle. Os efeitos heterogêneos observados indicam que programadores de pares com IA têm potencial para auxiliar na transição de pessoas para carreiras em desenvolvimento de software.

[Pina et al. 2022] investigaram problemas em repositórios de IA de código aberto para ajudar desenvolvedores a compreenderem dificuldades no uso de sistemas de IA. Foram analisados 576 repositórios da plataforma PapersWithCode, identificando 24.953 problemas com APIs REST do GitHub. O estudo categorizou os problemas em 13 tipos, sendo os mais comuns erros de tempo de execução (23,18%) e instruções pouco claras (19,53%). Observou-se que 67,5% dos problemas foram resolvidos, metade em até quatro dias. Funcionalidades de gerenciamento de problemas, como rotulagem e designação, são pouco usadas. O estudo recomenda o uso dessas funcionalidades e descrições detalhadas para melhorar a gestão e qualidade dos repositórios de IA de código aberto.

[Aghili et al. 2023] identificaram projetos AIOps no GitHub e analisaram métricas dos repositórios, como as linguagens de programação utilizadas. Em seguida, examinaram qualitativamente os projetos para entender os dados de entrada, as técnicas de análise e os objetivos. Por fim, avaliaram a qualidade dos projetos usando métricas como o número de bugs. Os resultados revelam um interesse recente e crescente em soluções AIOps. No entanto, a qualidade dos projetos AIOps é inferior à dos projetos usados como base. O estudo também identifica os problemas mais comuns nas abordagens AIOps e discute soluções potenciais.

5. Ameaças à validade

Nesta seção, são apresentadas as ameaças à validade deste estudo, assim como as estratégias adotadas para mitigá-las.

Primeiramente, quanto à validade de construção, o tamanho total da população de projetos Python no GitHub que dependem de bibliotecas ou frameworks relacionados à inteligência artificial não é conhecido. Para mitigar essa ameaça, foi utilizada uma quantidade relevante de repositórios.

Em relação à validade interna, a presença de dependência de uma biblioteca relacionada à IA nos arquivos *requirements.txt* não garante de fato que o projeto está relacionado à IA. Isso pode levar à inclusão de projetos que não utilizam de fato essas bibliotecas ou frameworks. Para diminuir essa ameaça, cada repositório Python foi clonado e todos os arquivos com a extensão .py foram verificados, buscando a confirmação das dependências através da análise dos *imports* no código.

Quanto à validade externa, a generalização dos dados coletados é limitada, pois os dados podem não refletir adequadamente todos os contextos e variações dos projetos, como diferenças entre projetos de código aberto e privado. Além disso, a diversidade nos métodos de gestão de dependências, incluindo o uso de LLMs e outras ferramentas emergentes, pode não ser totalmente representada. No entanto, essa ameaça é mitigada

pela abrangência da análise, que cobre um grande número de repositórios Python desenvolvidos por diversas organizações.

6. Conclusão

Neste estudo, foram analisados 15.770 repositórios Python entre os 100.000 mais populares do GitHub. Primeiramente, investigou-se quais bibliotecas e frameworks em Python para a área de inteligência artificial são mais utilizados. Em seguida, foi analisado o percentual de projetos Python que dependem de alguma biblioteca ou framework relacionado à inteligência artificial. Posteriormente, identificaram-se as características dos projetos Python que dependem dessas bibliotecas. Por fim, foram determinados quais países desenvolvem mais projetos em Python relacionados à inteligência artificial. Os resultados mostraram que as bibliotecas mais usadas foram TensorFlow, OpenCV e Scikit-learn. Observou-se também que 12% dos projetos Python analisados possuem pelo menos uma dependência de uma biblioteca ou framework relacionado à inteligência artificial. Além disso, verificou-se que esses projetos são relevantes, com uma mediana de 832 estrelas, 80 commits e 4 contribuidores. Constatou-se ainda que os países com mais projetos desenvolvidos são China, Estados Unidos e Alemanha. Como trabalhos futuros, pretende-se ampliar o estudo para outras linguagens de programação, realizar uma análise mais detalhada da qualidade do código e das práticas de manutenção nos projetos analisados, e incluir um estudo sobre o uso de LLMs (Modelos de Linguagem de Grande Escala) em projetos Python, ampliando o escopo além da verificação das dependências declaradas nos arquivos *requirements.txt* dos repositórios.

Pacote de Replicação

Os dados desta pesquisa estão disponíveis publicamente em https://doi.org/10.5281/zenodo.13344938.

Agradecimentos

Essa pesquisa é apoiada pelo IFPR.

References

Aghili, R., Li, H., and Khomh, F. (2023). Studying the characteristics of aiops projects on github. *Empirical Software Engineering*, 28(6):143.

Borges, H., Hora, A., and Valente, M. T. (2016). Understanding the factors that impact the popularity of GitHub repositories. In *32nd IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 334–344.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al. (2023). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.

Coelho, J. (2023). Crescendo, sobrevivendo ou morrendo? explorando a comunidade dos projetos brasileiros no github. In *Anais do XX Congresso Latino-Americano de Software Livre e Tecnologias Abertas*, pages 218–221. SBC.

Coelho, J., Valente, M. T., Milen, L., and Silva, L. L. (2020). Is this GitHub project maintained? measuring the level of maintenance activity of open-source projects. *Information and Software Technology*, 122:106274.

Coelho, J., Valente, M. T., Silva, L. L., and Shihab, E. (2018). Identifying unmaintained projects in GitHub. In *12th International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–10.

Dakhel, A. M., Majdinasab, V., Nikanjam, A., Khomh, F., Desmarais, M. C., and Jiang, Z. M. J. (2023). Github copilot ai pair programmer: Asset or liability? *Journal of Systems and Software*, 203:111734.

Fan, W., Zhao, Z., Li, J., Liu, Y., Mei, X., Wang, Y., Tang, J., and Li, Q. (2023). Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*.

Gomes, R. M. and Baunach, M. (2019). Code generation from formal models for automatic rtos portability. In *2019 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, pages 271–272. IEEE.

Gonzalez, D., Zimmermann, T., and Nagappan, N. (2020). The state of the ml-universe: 10 years of artificial intelligence & machine learning software development on github. In *Proceedings of the 17th International conference on mining software repositories*, pages 431–442.

Peng, S., Kalliamvakou, E., Cihon, P., and Demirer, M. (2023). The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590*.

Pina, D., Goldman, A., and Seaman, C. (2022). Sonarlizer xplorer: a tool to mine github projects and identify technical debt items using sonarqube. In *Proceedings of the International Conference on Technical Debt*, pages 71–75.

Shin, J. and Nam, J. (2021). A survey of automatic code generation from natural language. *Journal of Information Processing Systems*, 17(3):537–555.

Slowik, A. and Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications*, 32:12363–12379.

Tang, J. (2018). *Intelligent Mobile Projects with TensorFlow: Build 10+ Artificial Intelligence Apps Using TensorFlow Mobile and Lite for IOS, Android, and Raspberry Pi*. Packt Publishing Ltd.

Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., and Ting, D. S. W. (2023). Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Wong, M.-F., Guo, S., Hang, C.-N., Ho, S.-W., and Tan, C.-W. (2023). Natural language generation and understanding of big code for ai-assisted programming: A review. *Entropy*, 25(6):888.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.