A Study of Popular Artificial Intelligence Python Modules in Open Source Projects

Camila Reno[1], João Marcos Cardoso[1], Viviane Cordeiro[1], Paulo Meirelles[2], Phyllipe Francisco[1]

[1]Instituto de Matemática e Computação – Universidade Federal de Itajubá (UNIFEI)
37.500–903 – Itajubá – MG – Brasil

[2]Instituto de Matemática e Estatística (IME) – Universidade de São Paulo (USP)
Rua do Matão, 1010, Cidade Universitária, 05508-090 – São Paulo – SP

{camilamreno, joaomcoliveira, vivianecordeiro,phyllipe}@unifei.edu.br

paulormm@ime.usp.br

Abstract. *The increase in the use and popularity of Artificial Intelligence (AI) is directly related to the rise of the Python language, which is recognized for its simplicity and efficiency in AI projects. In this context, this study aimed to analyze the use of AI modules in public GitHub repositories. For this, the repository mining technique was applied using the Anonymous tool. A total of 142 popular repositories were analyzed, and we identified 17 that contained a set of predefined AI modules of interest. The most popular AI module identified was TensorFlow, present in just over 94% of these repositories. This highlights TensorFlow's dominance in AI projects, due to its robustness, active community, and integration with other tools. Furthermore, the predominance of this library reflects developers' preference for well-supported solutions with extensive practical applications. Our results complement the lists of popular libraries available online in grey literature, supporting professionals in making informed decisions when choosing libraries. They can align their projects with the most common and successful open-source practices.*

1. Introduction

Artificial Intelligence (AI) promotes opportunities and helps mitigate errors [Rodrigues and Andrade 2021]. Its impact can also be seen outside the tech industry, where it has generated a 20% financial gain for companies that adopted this type of technology [McKinsey & Company 2020]. This advancement is also seen in academia, where in 2015 the arXiv repository counted 5,478 publications related to AI. As of 2020, the number evolved to 34,736, highlighting a growing interest and development in the field [Gomes et al. 2021].

In this context, the Python programming language is widely recognized as one of the most important and used languages in developing AI solutions [Jenis et al. 2023], due to its versatility and the large number of available libraries. However, this diversity can present a challenge in identifying which tools are most suitable for developing AI solutions. Developers often face difficulties when choosing between libraries with similar functionalities but varying levels of support, documentation, and active community [Larios Vargas et al. 2020]. Knowing the most commonly used libraries can help

mitigate this problem, providing practical guidance based on empirical results obtained from repository mining [Ito et al. 2022]. Furthermore, the most popular modules or libraries can facilitate integration with existing projects, as they usually have a more active community, which suggests richer documentation and broader support. Additionally, this scenario can minimize rework and risks associated with choosing poorly supported libraries [Larios Vargas et al. 2020, Nguyen et al. 2020].

To address the challenges related to the choice and use of AI libraries in Python, this study proposes a mapping of the most commonly used libraries in open-source projects. The methodology combines repository mining techniques and exploratory data analysis to identify the most popular modules and the software domains in which they are used. To guide our research, we elaborated the following questions:

- **RQ1**: What are the most popular modules in open-source projects?
- **RQ2**: What are the most common AI modules used in open-source projects?
- **RQ3**: Which AI modules are most frequently used in combination?
- **RQ4**: What software domains are the most common in open-source projects using AI modules?

For the study, 142 projects were selected using criteria such as GitHub stars and excluding academic/tutorial projects. Next, a list of AI-specific modules of interest (described in Section 3) was compiled. Among the results, TensorFlow is the most widely used library in AI-related repositories. In terms of module combinations, the use of TensorFlow and Keras is the most common.

2. Python Programming Language

Python is a popular choice in various fields, including web development, data science, automation, and artificial intelligence. The community has played an important role in popularizing and growing the language, especially in the context of artificial intelligence projects. According to GitHub's 2020 report [GitHub 2020], the language's growth also reflects the expansion of a community of professional or amateur data scientists [Raschka et al. 2020]. In addition, the community promotes collaboration and knowledge sharing, facilitating access to educational resources, tutorials, and technical support. This process accelerates innovation and helps Python remain the preferred language for AI development [Jenis et al. 2023].

Another fundamental component of the Python language is modules. These allow for efficient code organization and reuse [Python Software Foundation 2024]. They make it easier to create libraries that can be imported and applied to different parts of a project or across multiple projects. Python's modularity simplifies development and improves code maintenance and scalability. We highlight that the terms "module" and "library" have conceptual differences [Foundation 2024a], but they are being used interchangeably in this work.

The vast collection of available libraries, such as NumPy [Harris et al. 2020] for advanced mathematical operations and TensorFlow [Abadi et al. 2015] for AI development, exemplifies the power of modularity. They offer optimized solutions and enable collaborative development in open-source projects, accelerating advances in fields such as machine learning and data science. The modularity and integration capability of Python

modules significantly contribute to its popularity in AI, solidifying its role as one of the main programming languages in the field [Joshi and Tiwari 2023].

By identifying the most popular libraries and how they are used in real-world projects, this research aims to support developers in selecting libraries for their projects. Additionally, it helps provide a better understanding of industry demands, contributing to the alignment of candidate and employee training and offering valuable insights for companies, such as startups.

3. Methodology

This section presents the research design adopted to reach the main goal of this study, i.e., understanding and analyzing the most commonly used Python modules for artificial intelligence in open-source projects. The methodology was based on the studies of [Lima et al. 2018, Tutko et al. 2022]. The following is a list of actions performed to reach our main goal: (i) Research Questions, (ii) PySniffer Tool, (iii) Project Selection, (iv) Libraries of Interest, and (v) Exploratory Data Analysis. A replication package is available[1].

3.1. Research Questions

As discussed in Section 2, Python's popularity is closely tied to relevant AI libraries. This study aims to identify these modules and how the language is being adopted in the rapidly growing AI field [Saabith et al. 2020]. The following questions guide our research:

- Q1 - What are the most popular modules in open-source projects? This question seeks to identify the most commonly used Python libraries in public repositories, specifically on GitHub.
- Q2 - What are the most common AI modules used in open-source projects? This question aims to identify the most reliable and popular libraries for AI applications and highlight. It also aims to identify what software domains are associated with each module.
- Q3 - Which AI modules are most frequently used in combination? This question explores usage patterns among specific modules – for example, deep learning packages combined with data handling or visualization libraries.
- Q4 - What software domains are the most common in open-source projects using AI modules? This question aims to understand which software domains appear most frequently in the analyzed repositories. We will elaborate on a list of software categories to answer this question. This analysis may aid in identifying trends related to specific libraries.

3.2. PySniffer

The PySniffer tool was first developed in the work of [Ito et al. 2022]. This tool can scan Python projects, extract the imported modules from each ".py" file, and generate a JSON. The authors also built a front-end interface for easier data analysis. We used this tool and adapted it to meet our research needs. In short, we added the following features:

- Filtering of selected libraries

[1] https://doi.org/10.5281/zenodo.16773496

- Counting the number of files in each repository

PySniffer uses the standard library module `ast` to identify import statements in ".py" files, which builds an Abstract Syntax Tree. This allows the tool to parse the code and locate import lines. The `pipreqs` module is also used to determine which libraries are used in each project and identify the most frequent ones. The modified version we developed for this work can be found in:
`https://github.com/Joao-MCO/PySniffer/tree/unifei-tcc`.

Figure 1 shows the basic flow of PySniffer, including the custom modifications we made for this study.

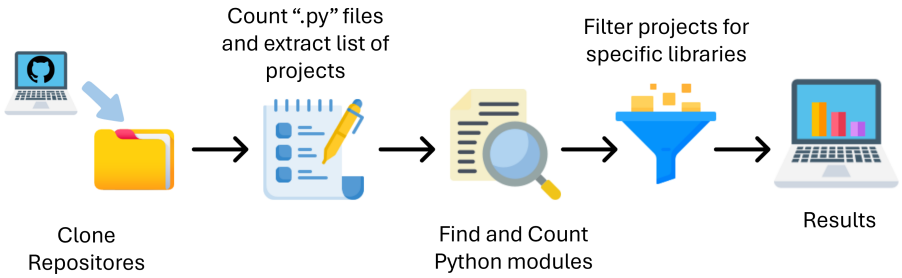

Figure 1. General flow of the PySniffer Tool

3.3. Project Selection

The used data consists of open-source projects hosted on GitHub. A total of 142 projects were manually selected based on the following criteria:

- Written in Python;
- More than 20,000 stars;
- Must not be collections of Python examples or course/book scripts;
- Not the official CPython repository;
- English README.

[Borges and Tulio Valente 2018] explains that GitHub stars reflect repository popularity. Projects with more stars are considered more influential and relevant. The 20,000 star threshold was chosen to narrow down the sample size to a manageable level. As a comparison, the work of [Ito et al. 2022] analyzed 129 Python repositories.

3.4. Libraries of Interest

To specifically search for projects using AI modules, we elaborated a list analysing the work of [Sundaram et al. 2023], [Joshi and Tiwari 2023], and [Haoran 2022]. The authors cite libraries used for AI applications in the context of Python. The PySniffer tool will select, among the 142 projects, which ones contain these AI libraries. Afterward, we can find the most popular ones and categorize their software domains. This list is on Table 1.

Table 1. Selected Artificial Intelligence Modules

TensorFlow	PyTorch
OpenCV	Caffe
Keras	MXNet
Scikit-learn	CNTK
Nolearn	Theano
PySpark	TFLearn
Statsmodels	Lasagne
PyCaret	Elephas
Shogun	MLpy

3.5. Exploratory Data Analysis

We conducted an exploratory data analysis on the selected projects' most commonly used Python AI modules. We then categorized these projects to help identify software application domains. For instance: frameworks, image recognition, chatbot, etc. The authors manually reviewed each repository by analyzing its README file, which typically outlines the project's goals and scope. The results from this analysis were used to answer the research questions, guiding the study toward its objectives.

4. Experiments and Results

This section presents the results and analysis of our experiments, obtained from extracting the AI modules from the selected 142 repositories. The results are organized into four subsections, each addressing one of the proposed questions.

4.1. The Most Popular Modules in Open Source Projects

From the analysis of the 142 projects selected for this research, 98,174 Python files were identified. With the execution of the PySniffer tool, a total of 2,547 modules were found. It was observed that most of these modules, 2,347 (92%), correspond to third-party libraries, while 200 modules belong to the standard library.

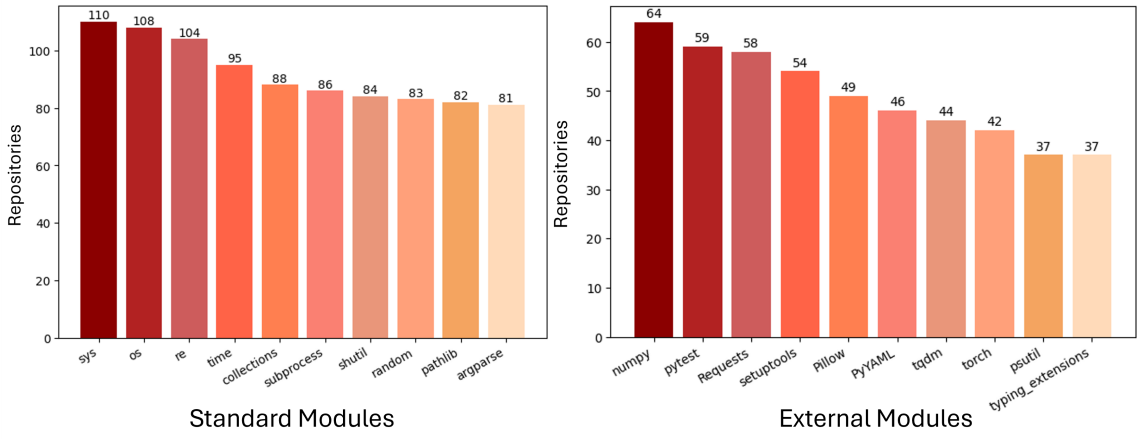

Figure 2. Standard and External Modules

Table 2. Description of the most popular standard modules

Module	Description
sys	Accesses interpreter variables and functions
os	Handles files, directories, and system processes
re	Handles text using regular expressions
time	Works with time measurements and date/time manipulation
collections	Implements specialized container data types that provide alternatives to built-in Python containers
subprocess	Executes and controls OS processes
shutil	Handles file and directory operations, including copying and deletion
random	Generates random numbers and performs random operations
pathlib	Manages file and directory paths using an object-oriented approach
argparse	Creates and processes command-line arguments

Table 3. Description of the most popular external modules

Module	Description
numpy	Library for efficient operations with arrays and linear algebra
pytest	Framework for writing and running automated tests
requests	Performs simple and flexible HTTP requests
setuptools	Manages packages and dependencies for Python projects
pillow	Processes and manipulates images in various formats
pyYAML	Reads and writes YAML files
tqdm	Displays progress bars for loops and long-running processes
torch	Library for tensor computation and deep learning
typing extensions	Provides advanced typing features for backward compatibility with older Python versions
psutil	Monitors and collects information about system processes and usage

The most popular standard and external modules are shown in Figure 2. Tables 2 and 3 shows, respectively, the description of the funcionalites of the main and external modules.

These results highlight the importance of specific libraries to the Python language and their connection to the AI field. For example, `NumPy` is a fundamental tool for mathematical and computational operations, widely used in data science and machine learning due to its efficient handling of arrays and linear algebra. Libraries like `TQDM`, often used to monitor the progress of long-running tasks, are relevant in AI model training contexts. The presence of `Torch` directly reflects the growing popularity of deep learning, showing significant adoption in open-source projects. Other trends also emerge from these results, such as `Requests`, highlighting HTTP usage; `Pytest`, reflecting code quality and test automation concerns; and `Pillow` and `PyYAML`, emphasizing image processing and structured data handling.

4.2. The Most Common AI Modules in Open Source Projects

Considering only the modules of interest listed in Table 1, the PySniffer tool identified the most common AI modules in open-source projects. Of the 142 initially analyzed projects, 17 contain libraries of interest. Figure 3 shows a list of five modules in these projects and

the number of occurrences for each.

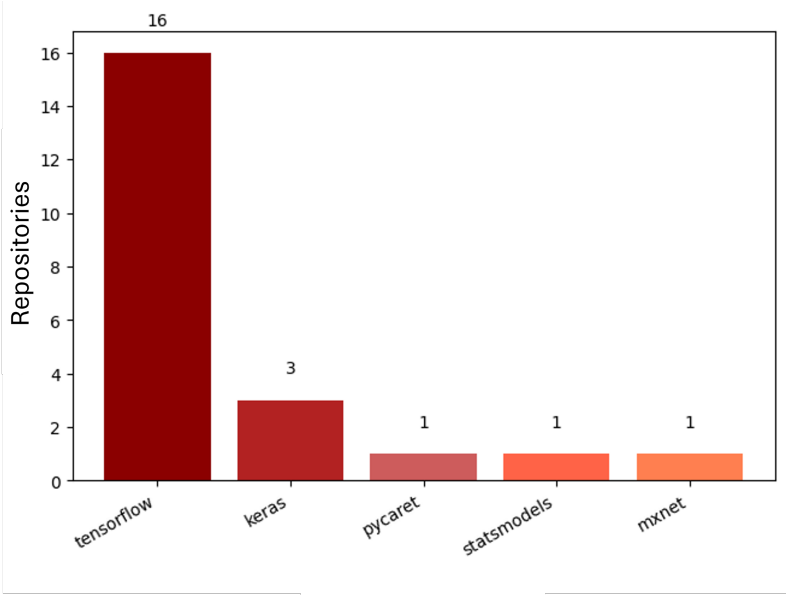

Figure 3. AI modules.

According to the chart, TensorFlow was the most frequently found library in 16 of the 17 analyzed projects. It was followed by Keras in 3 projects, then PyCaret [Ali 2020], Statsmodels [Community 2024], and MXNet [Foundation 2024b].

TensorFlow, developed by Google and widely used for machine learning and neural network models, offers functionalities for training deep neural networks such as convolutional and recurrent networks, often used in speech recognition and natural language processing [TensorFlow 2024b, TensorFlow 2024c]. Since version 2.0, Keras has been integrated into TensorFlow as its official high-level API, simplifying neural network construction [TensorFlow 2024a]. Keras can still be used independently with other packages, offering intuitive APIs for deep learning model creation.

PyCaret is notable for automating machine learning workflows with minimal coding [Ali 2020], handling tasks like data preprocessing and hyperparameter tuning. Statsmodels focuses on statistical analysis and modeling. It offers linear regression, time series models, and statistical tests. MXNet, developed by the Apache Software Foundation, is designed for scalability, enabling model training on distributed computing systems like machine clusters [Foundation 2024b], supporting both imperative and declarative neural network programming.

4.3. The Most Frequently Combined AI Modules

This study also aimed to identify which AI modules are most often used in combination. Table 4 shows the combined modules and the repositories where they were found. TensorFlow and Keras stood out as the most frequent combination.

As discussed in Section 4.2, Keras became TensorFlow's official interface, which explains their pairing. Developers often prefer this combination for simplifying model development and reducing coding time, as Keras provides a user-friendly interface while

Table 4. Modules found in combination

Repositories	Modules Found
Faceswap	Keras, TensorFlow
Roop	Keras, TensorFlow
Mask RCNN	Keras, TensorFlow
Mindsdb	Pycaret, TensorFlow
Pytorch-image-models	Mxnet, TensorFlow

TensorFlow handles the backend complexity. The PyCaret–TensorFlow combination suggests that developers seek to automate and streamline AI model training.

The use of MXNet alongside TensorFlow, as in the "Pytorch-image-models" project, illustrates developers' flexibility in adopting different libraries for specific tasks. MXNet's scalability makes it ideal for performance-critical, distributed computing environments.

4.4. Popular Categories Among Projects of Interest

To categorize these projects the authors performed a qualitative analysis by reading each README file. These provided data that helped us elaborate the following list of software application domains:

- **Image Modeling:** Repositories focused on image manipulation, such as face swapping, real-time effects, and visual transformations.
- **Detection and Tracking:** Projects focused on computer vision to identify objects in images/videos, used in security and traffic analysis.
- **Web Application:** Web-based interfaces for computer vision tasks like object detection and tracking.
- **Framework:** Toolsets and patterns that ease application development.
- **Recommendation:** Projects using machine learning to make predictions and suggestions from data.
- **Finance:** Tools for financial analysis and investment strategies.
- **Driver Assistance:** Autopilot systems contributing to autonomous driving.
- **Chatbot:** Virtual assistants focused on natural language interaction.
- **Models:** Pre-trained models for computer vision tasks.
- **Utility:** Projects that add progress bars to code loops.
- **Audio Management:** Tools for separating audio sources (e.g., vocals/instrumentals).
- **Natural Language Processing (NLP):** Deep learning models for tasks like translation, summarization, and sentiment analysis.

Table 5 presents the relation of the projects and their categories.

Two categories lead in popularity with three projects each: "Image Modeling" and "Detection and Tracking". These areas are commonly applied to visual effects, video editing, live streaming, security, industrial inspection, and traffic monitoring. Two projects are categorized under Web Application, such as Gradio and Streamlit, used to create interactive ML interfaces. The rest fall under distinct, non-repeating categories.

Table 5. Repositories Grouped by Category

Category	Repositories
Image Modeling	Faceswap Deep-Live-Cam Roop
Detection and Tracking	Ultralytics Yolov5 Mask RCNN
Web Application	Streamlit Gradio
Framework	Pytorch
Recommendation	Mindsdb
Finance	OpenBB
Driver Assistance	Openpilot
Chatbot	Open-Assistant
Models	Pytorch-image-models
Utility	Tqdm
Audio Management	Spleeter
NLP	Bert

The six leading projects from the most popular categories focus on image manipulation. They represent 35.2% of all analyzed projects. The chart in Figure 4 shows the categories' distribution and project percentages.

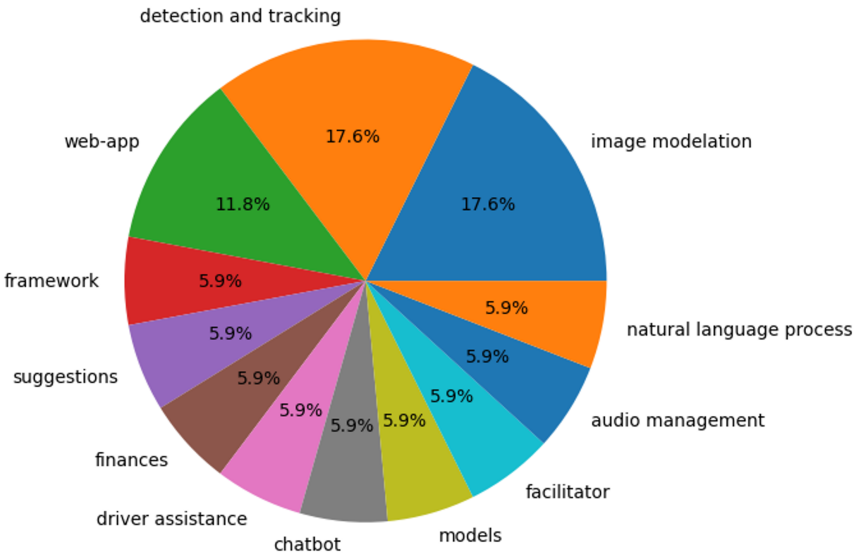

Figure 4. Classification of AI-focused repositories.

5. Conclusion

This work conducted a study to find the most widely used Python modules and, more specifically, Artificial Intelligence (AI) modules in open-source projects hosted on the GitHub platform. This list can aid developers when choosing libraries/modules with similar functionalities for their projects. We selected 142 projects, totaling 98,174 Python

files. Among these, 17 projects were identified as being of interest, i.e., meaning their solutions were directed toward the AI field.

The results revealed that, among the external libraries, `NumPy`, `Pytest`, and `Requests` were the most used, while the most frequent standard modules were `Sys`, `Os`, and `Re`. Considering the 17 projects of interest, `TensorFlow` was identified as the most commonly used module, followed by `Keras`, `PyCaret`, `Statsmodels`, and `MXNet`. The most common combinations showed that `TensorFlow` and `Keras` frequently coexist in the development of the same project.

One threat to the validity of the results obtained in this work is the criteria used for repository selection. Although these were detailed, it is possible to obtain a different list of projects than the one obtained in this research. Repositories may change daily, either in the number of stars or their functionalities.

As future work, we intend to perform a historical analysis of the libraries, seeking to identify popularity trends over time. This approach will allow us to understand which libraries stood out in specific periods and how the use of these technologies evolved, enabling developers to visualize new scenarios in choosing solutions that are more suitable for their needs.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Ali, M. (2020). Pycaret: An open source, low-code machine learning library. Disponível em `https://pycaret.org/`, acessado em 1 de dezembro de 2024.

Borges, H. and Tulio Valente, M. (2018). What's in a github star? understanding repository starring practices in a social coding platform. *Journal of Systems and Software*, 146:112–129.

Community, S. (2024). Statsmodels: Statistical models in python. Acesso em: 01 dez. 2024.

Foundation, P. S. (2024a). Python modules tutorial. Acesso em: 15 dez. 2024.

Foundation, T. A. S. (2024b). *Apache MXNet: A Flexible and Efficient Deep Learning Framework*. The Apache Software Foundation. Versão 1.9.1.

GitHub (2020). The state of the Octoverse 2020.

Gomes, L. I. E., Fernández Marcial, V., and Santos, M. N. (2021). O impacto da inteligência artificial nos serviços de informação: inovação e perspetivas para as bibliotecas. In *Organização do Conhecimento no Horizonte 2030: Desenvolvimento Sustentável e Saúde: Atas do V Congresso ISKO Espanha-Portugal.*, pages 393–405. Centro de Estudos Clássicos, Colibri.

Haoran, X. W. . Y. (2022). Python libraries for data analysis and machine learning. Accessed: 2024-10-25.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Ito, L. G., Moreira, M. H. I., Souza, S. B., Medeiros, S. P., and Lima, P. (2022). What are the top used modules in python open-source projects? *Anais do Computer on the Beach*, 13:037–044.

Jenis, J., Ondriga, J., Hrcek, S., Brumercik, F., Cuchor, M., and Sadovsky, E. (2023). Engineering applications of artificial intelligence in mechanical design and optimization. *Machines*, 11(6).

Joshi, A. and Tiwari, H. (2023). An overview of python libraries for data science: Manuscript received: 20 march 2023, accepted: 12 may 2023, published: 15 september 2023, orcid: 0000-0003-0873-3340, https://doi. org/10.33093/jetap. 2023.5. 2.10. *Journal of Engineering Technology and Applied Physics*, 5(2):85–90.

Larios Vargas, E., Aniche, M., Treude, C., Bruntink, M., and Gousios, G. (2020). Selecting third-party libraries: the practitioners' perspective. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2020, page 245–256, New York, NY, USA. Association for Computing Machinery.

Lima, P., Guerra, E., Meirelles, P., Kanashiro, L., Silva, H., and Silveira, F. (2018). A metrics suite for code annotation assessment. *Journal of Systems and Software*, 137:163–183.

McKinsey & Company (2020). The State of AI in 2020. Acesso em: 26 nov. 2024.

Nguyen, P. T., Di Rocco, J., Di Ruscio, D., and Di Penta, M. (2020). Crossrec: Supporting software developers by recommending third-party libraries. *Journal of Systems and Software*, 161:110460.

Python Software Foundation (2024). The python tutorial: Modules. `https://docs.python.org/3/tutorial/modules.html`. Acessado: 17-ago-2024.

Raschka, S., Patterson, J., and Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence.

Rodrigues, B. and Andrade, A. (2021). O potencial da inteligência artificial para o desenvolvimento e competitividade das empresas: uma scoping review. *Gestão e Desenvolvimento*, (29):381–422.

Saabith, A. S., Vinothraj, T., and Fareez, M. (2020). Popular python libraries and their application domains. *International Journal of Advance Engineering and Research Development*, 7(11).

Sundaram, J., Gowri, K., Devaraju, S., Gokuldev, S., Jayaprakash, S., Anandaram, H., Manivasagan, C., and Thenmozhi, M. (2023). An exploration of python libraries in machine learning models for data science. In *Advanced Interdisciplinary Applications of Machine Learning Python Libraries for Data Science*, pages 1–31. IGI Global.

TensorFlow (2024a). Keras: A high-level api for tensorflow. `https://www.tensorflow.org/guide/keras`. Acessado em: 12 dez. 2024.

TensorFlow (2024b). Tensorflow core: Convolutional neural networks. `https://www.tensorflow.org/tutorials/images/cnn`. Acessado em: 12 dez. 2024.

TensorFlow (2024c). Tensorflow core: Working with rnns. `https://www.tensorflow.org/guide/keras/rnn`. Acessado em: 12 dez. 2024.

Tutko, A., Henley, A. Z., and Mockus, A. (2022). How are software repositories mined? a systematic literature review of workflows, methodologies, reproducibility, and tools. *arXiv preprint arXiv:2204.08108*.