

LLMs Love Python: Strong Biases For Coding Languages & Libraries Pose Hidden Risks to Software Quality

Lukas Twist, Jie Zhang, Mark Harman, Don Syme, Joost Noppen, Detlef Nauck

Background:

- LLMs are contributing to more code than ever before, but the decision-making process behind their **coding language and library selection remains unexplored**.
- This is particularly significant as LLMs are increasingly used in **real-world software development workflows**.
- If LLMs exhibit systematic biases or inconsistent recommendations, it may lead to flawed decisions by developers, **compromising software reliability, security, or maintainability**.

Method:

- Empirical study across 8 LLMs including ChatGPT, Claude, Llama, DeepSeek, & Mistral.
- Find their preferences when the technology is not specified.
- Do this for both solving **benchmark problems**, and initialising **new project code**.

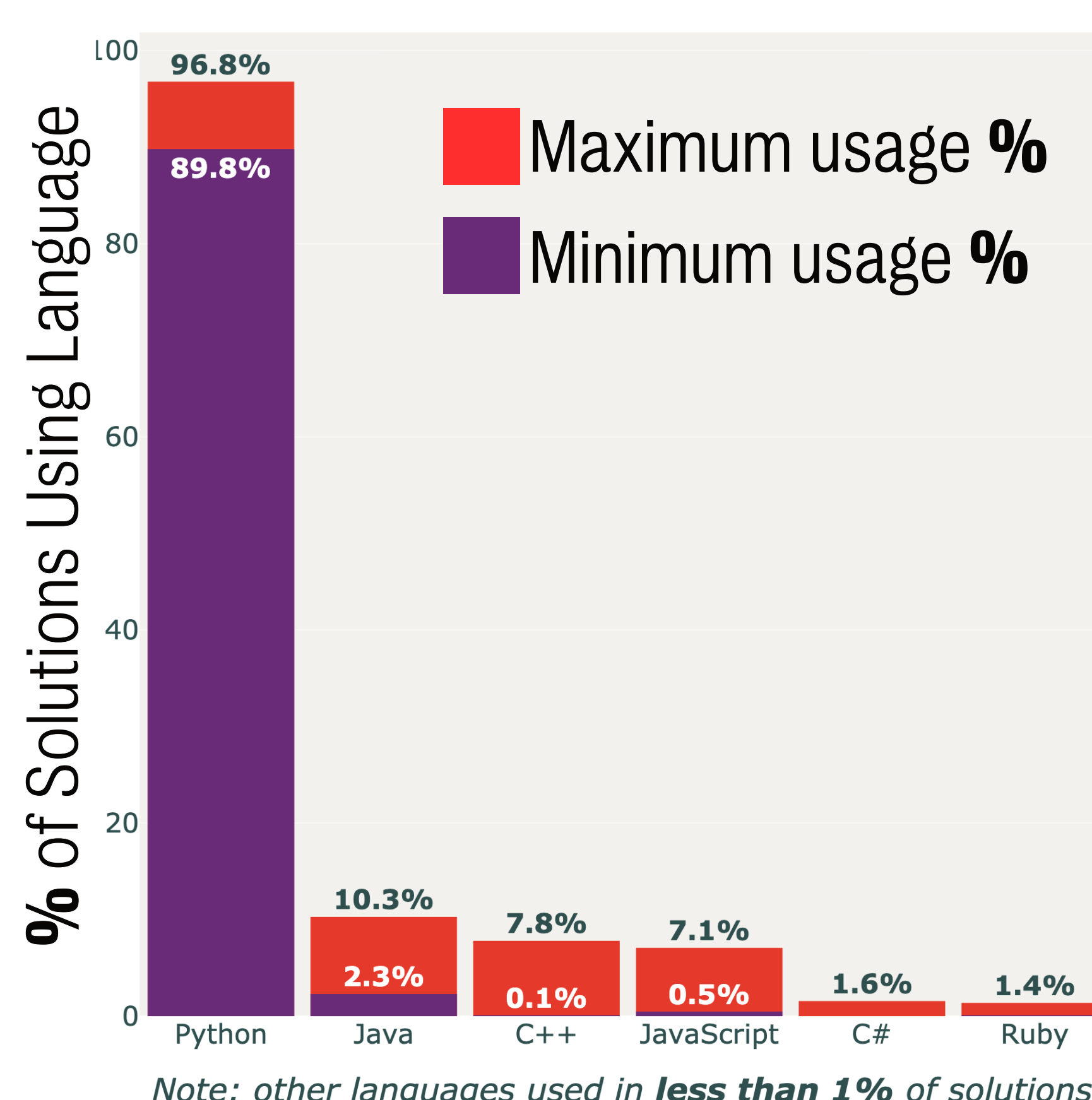
Results for Language Preferences:

- LLMs use **Python** to solve benchmark problems **90-97%** of the time.
- When generating initial project code where **Python is not suitable**, it is the **most-used language 58%** of the time.
- In general, LLMs **contradict** their own language recommendations in **83%** of project initialisation tasks.

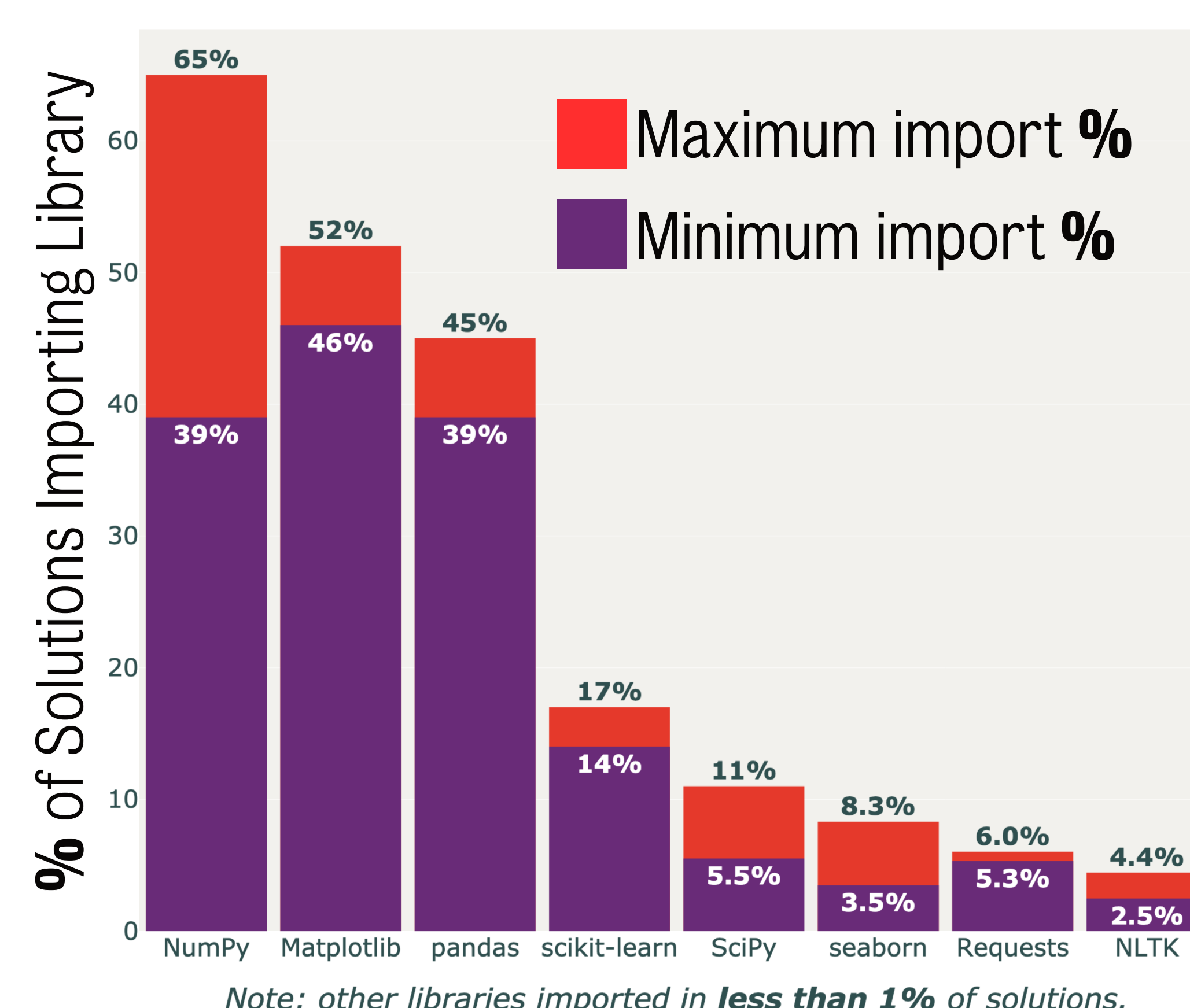
Results for Library Preferences:

- LLMs **heavily favour top-ranked libraries** such as NumPy, yet in **up to 48%** of cases this usage is **unnecessary** and differs from the ground truth.
- In general, they use a limited range of **32-39 libraries** for benchmark tasks.
- When starting new projects, LLMs strongly **favour older, established** libraries over **new, higher quality** options (according to GitHub metrics).

Languages used for benchmark tasks:



Libraries used for benchmark tasks:



Key Takeaways:

- Strong biases are seen across **all LLMs** in the study.
- Favouring mainstream technologies **limits LLM effectiveness** when producing more **specialised code**.
- As LLMs contribute to more code, the coding landscape will begin to **converge on their preferences**.
- This lack of diversity also means they will rarely use new technologies, causing a **discoverability problem for open-source projects**.

