Observational study on LLM technology usage

(In public GitHub repositories)

Goal Question and Metrics

We want to respond to the following LLM usage and popularity related questions with our research:

- How has the popularity of LLM models evolved since 2022?
- What is the most popular language for LLM software development?
- What is the most used SDK for this purpose?
- What is the most popular LLM model used for inference?
- How are these models being used in practice?
- How do LLM projects in different languages compare to each other?

Goal Question and Metrics

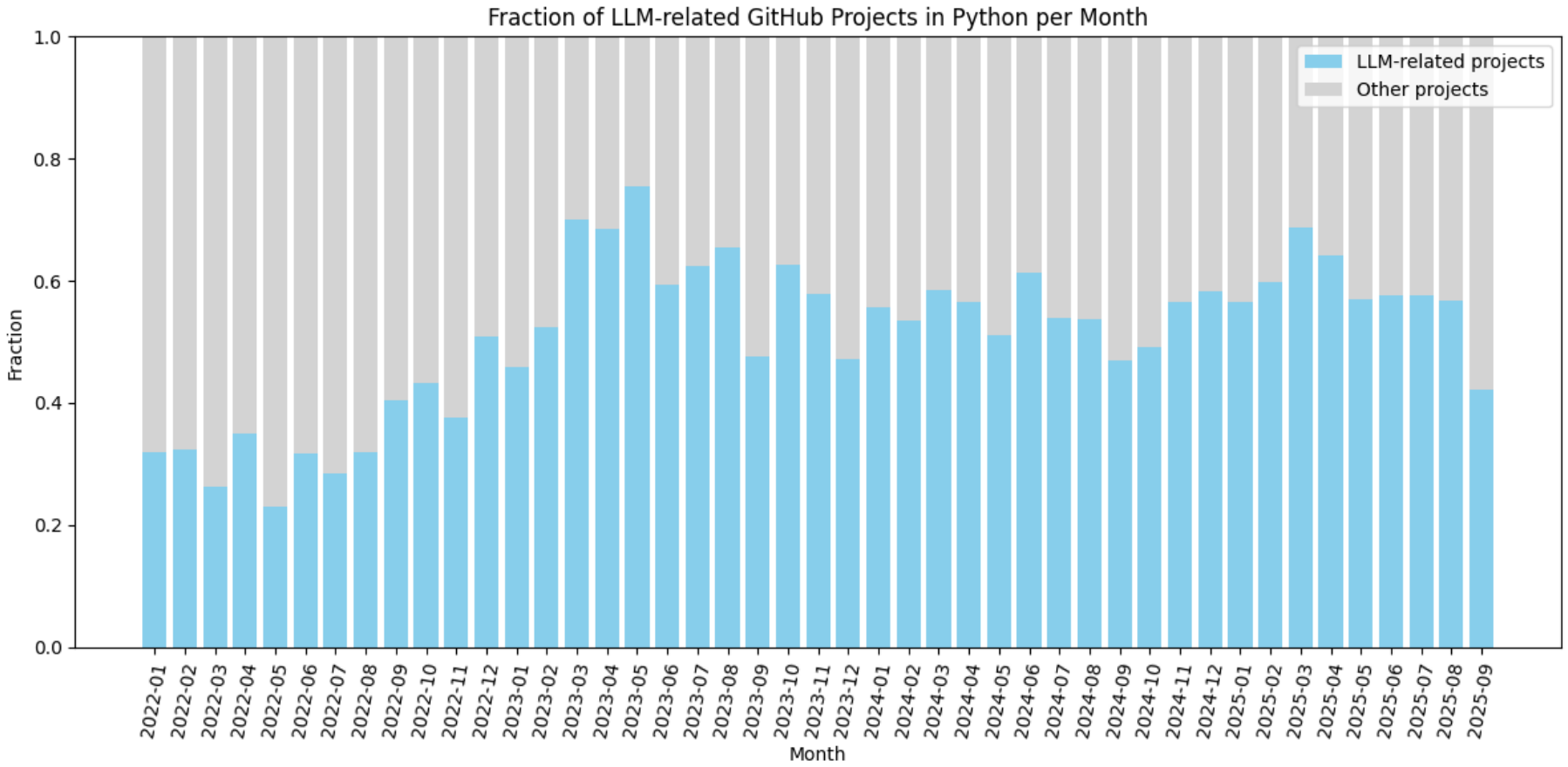
To achieve these objectives, we have divided our experiment into three major sections:

- Monthly most popular LLM-related repositories (with keyword-search)
- Exhaustive search of LLM-related repositories (with model and library-search)
- Random sampling of LLM-related repositories (with attribute search)

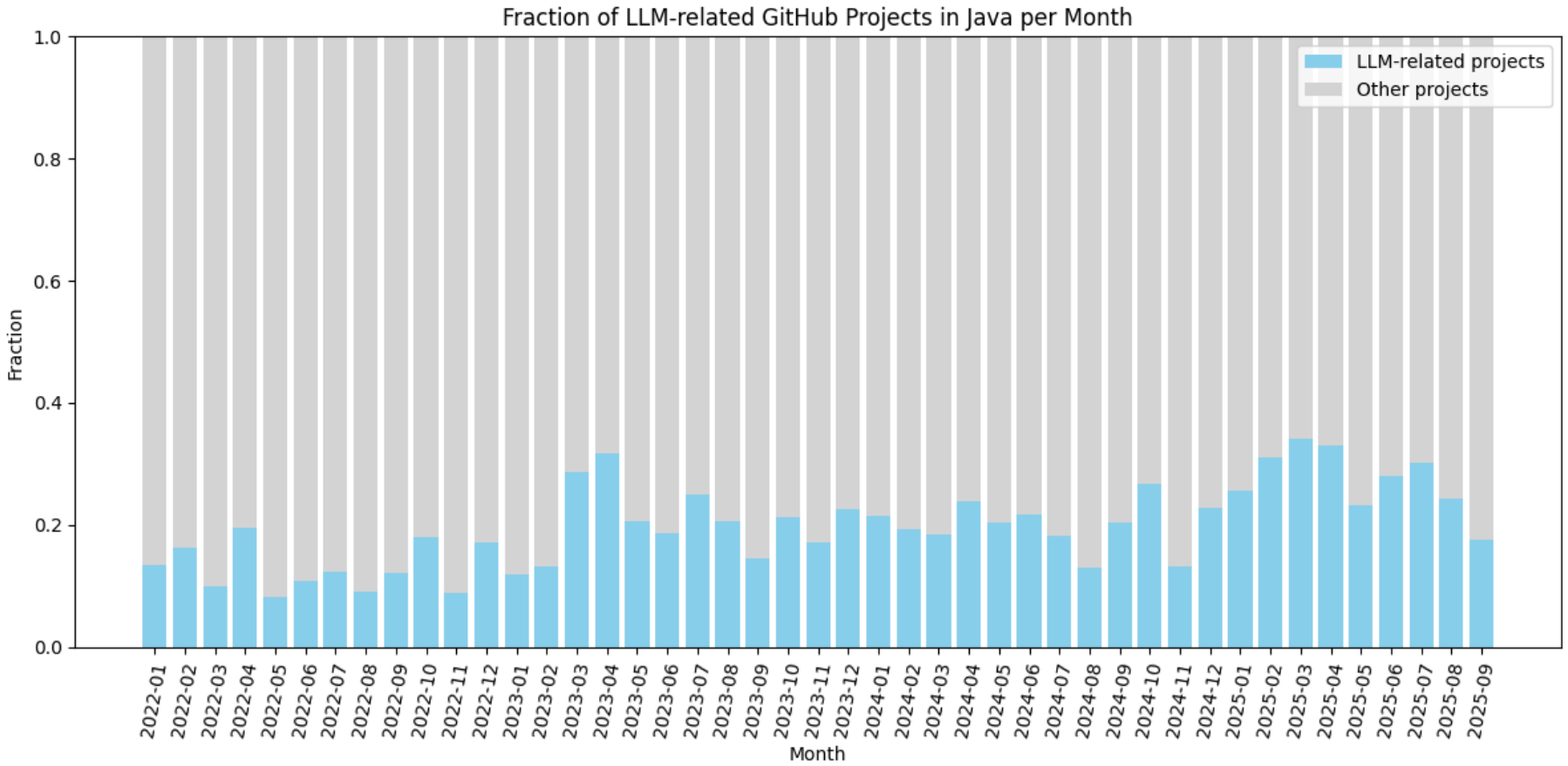
Keyword Search

An analysis of the top 100 most popular LLM-related repositories by matching LLM keywords to a repository's name, description, or topics. Only repositories created between 2022 and October 2025 were considered.

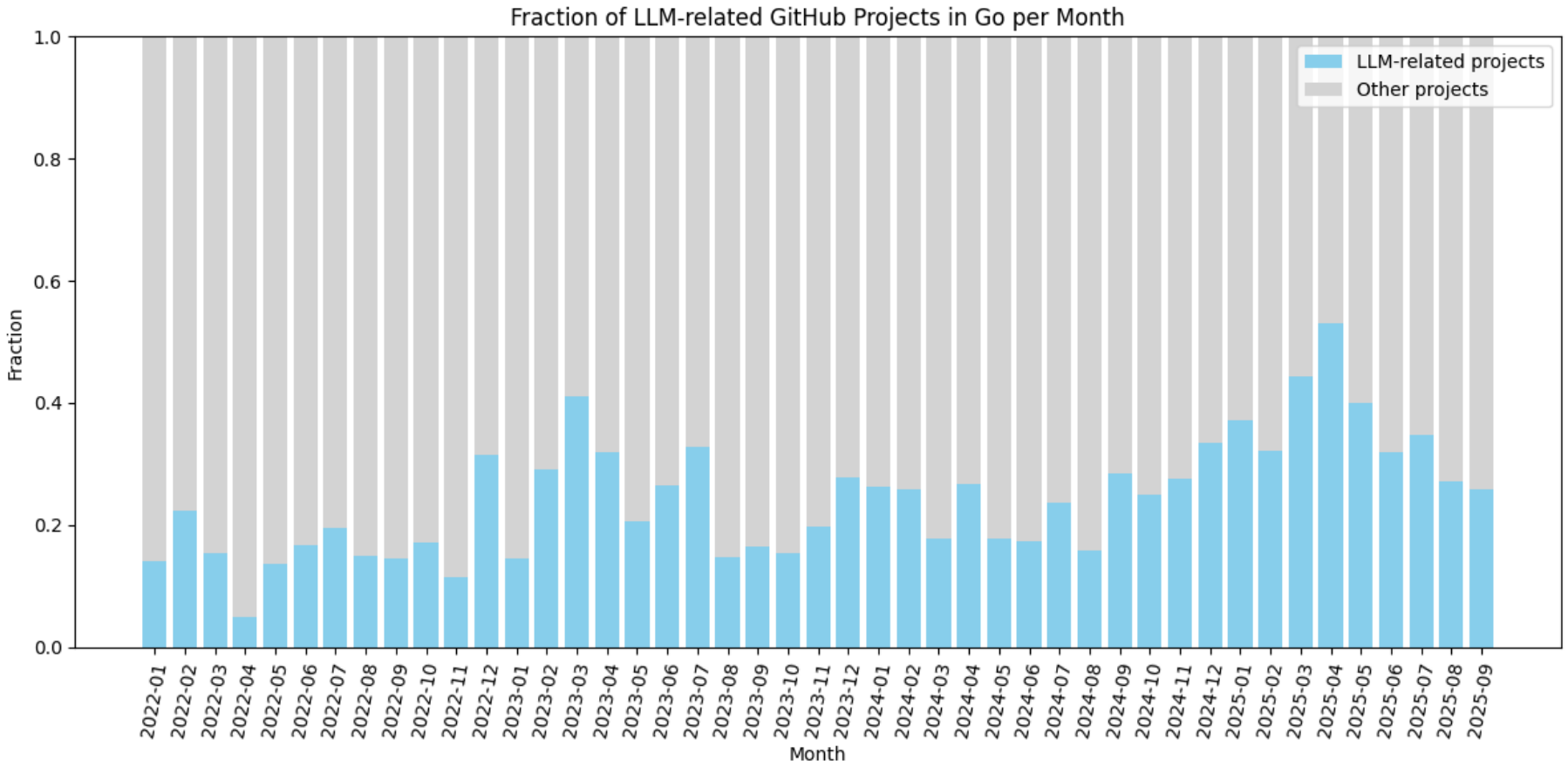
Keyword Search – Python LLM Popularity evolution



Keyword Search – Java LLM Popularity evolution



Keyword Search – Go LLM Popularity evolution



Library Search

At this stage of the experiment, we try to recover as many LLM-related repositories as possible using GitHub's code search endpoint and employing the chunking technique. In particular, here we search for repositories containing imports of notable LLM SDKs.

Model Search

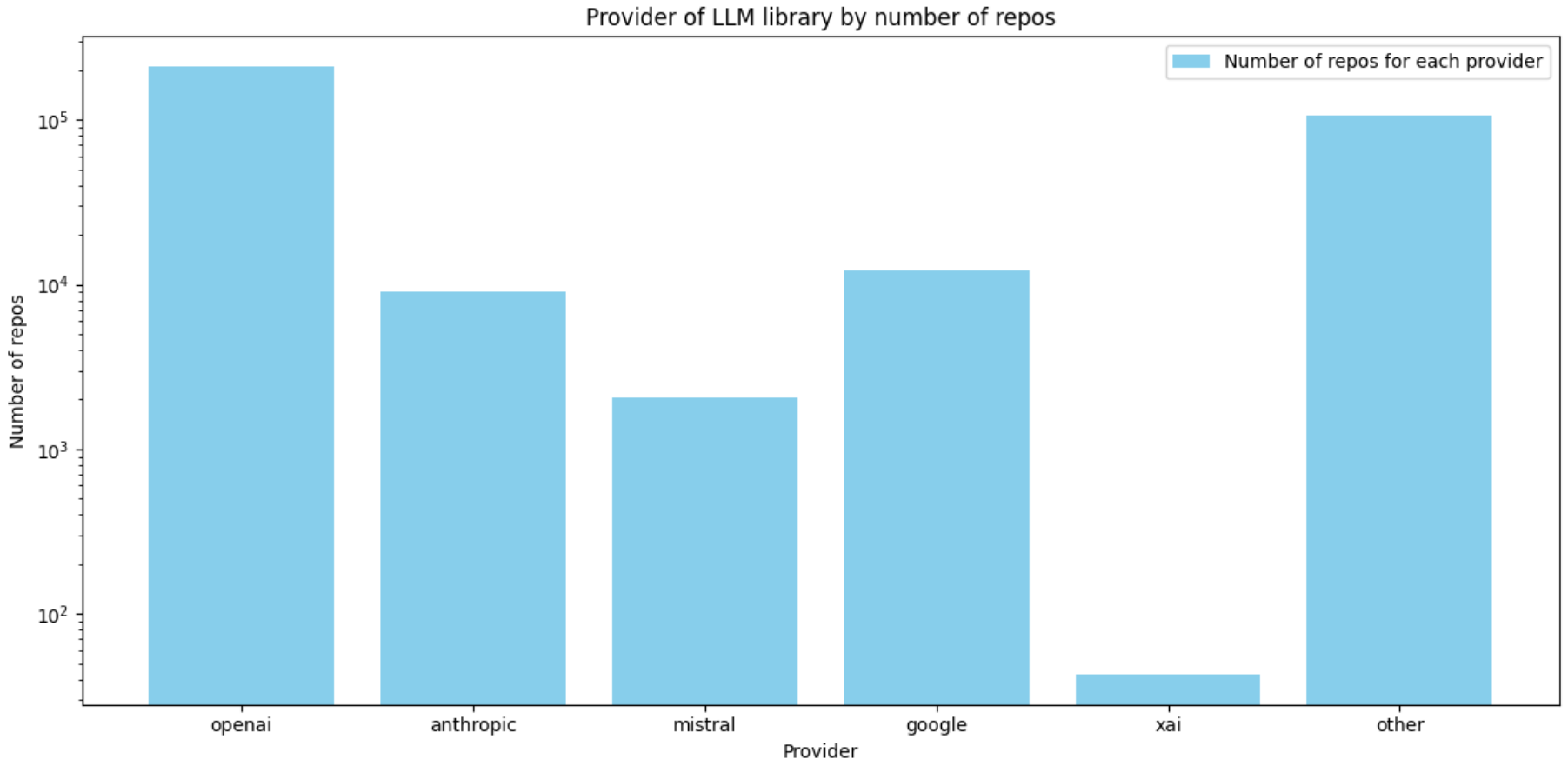
Very similar to Library Search, but instead of searching for notable library imports, we search for notable LLM model names. The results from the two approaches are then combined to identify correlations between the two.

Library & Model Search: Popularity of languages for LLM software development

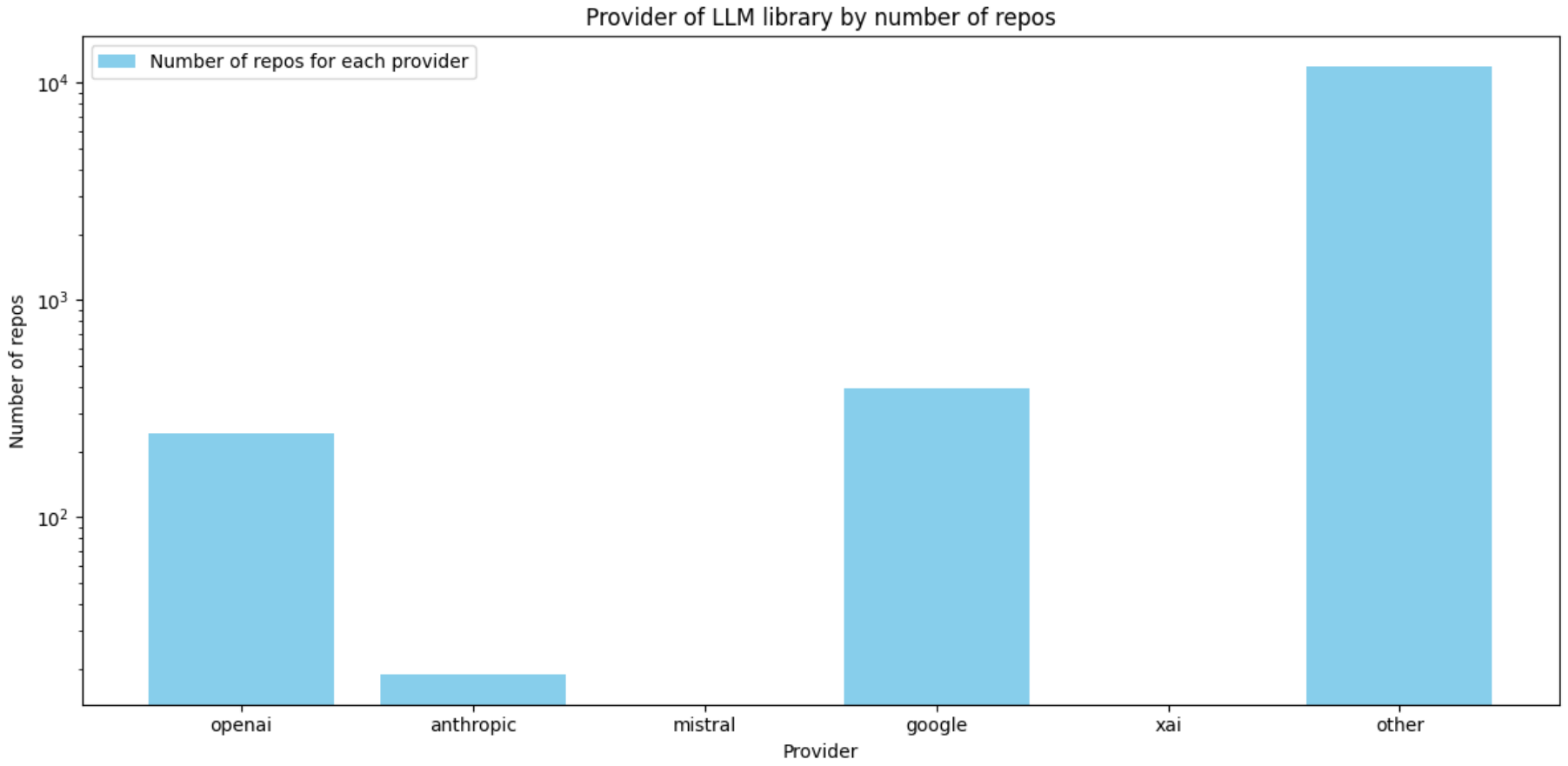
To evaluate the popularity of each language for LLM-integrated software development, we simply measure the number of repositories we found in each combined file:

- For Python: 262581
- For Java: 8639
- For Go: 5511

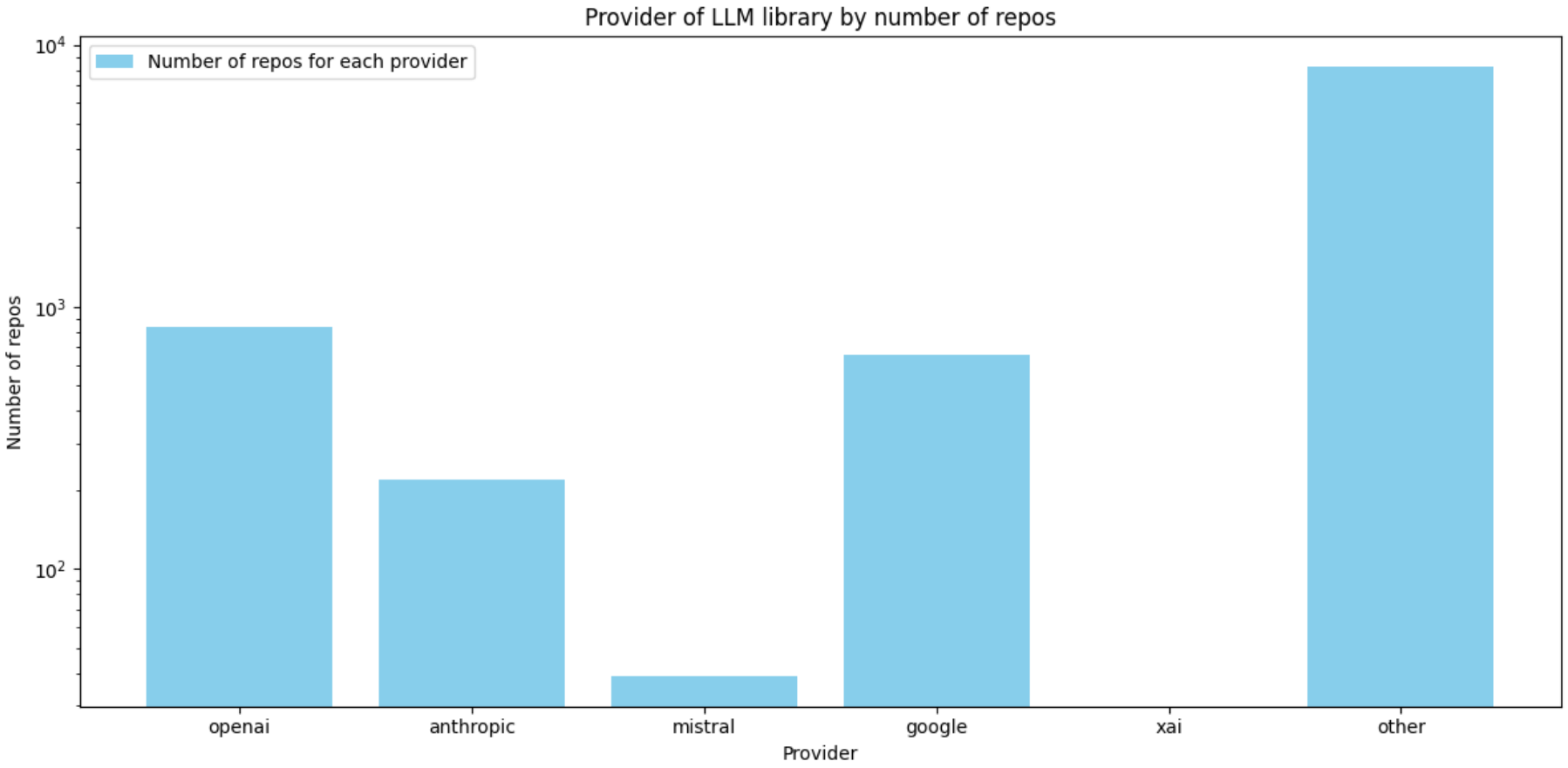
Library & Model Search – Python LLM SDK popularity



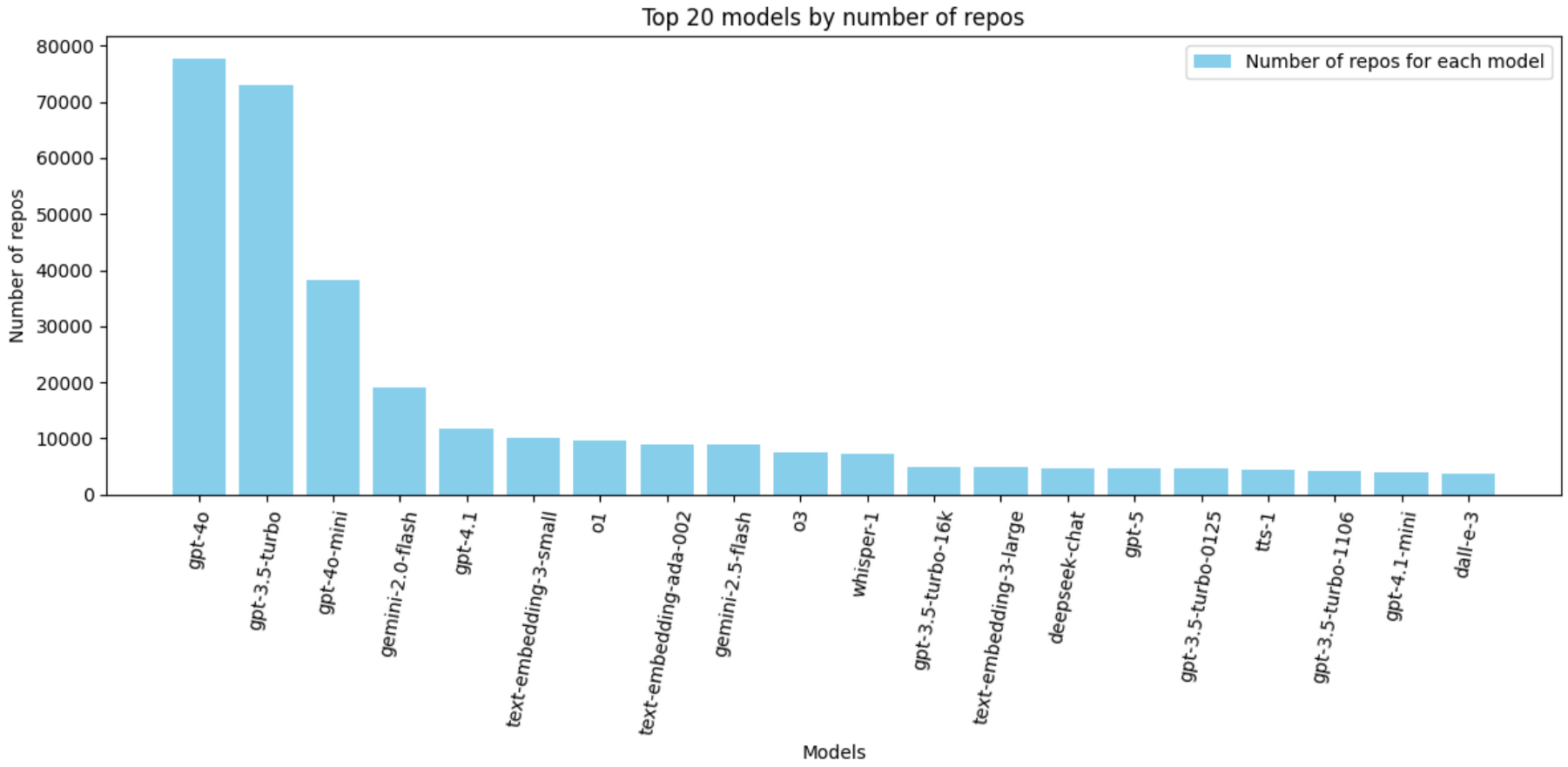
Library & Model Search – Java LLM SDK popularity



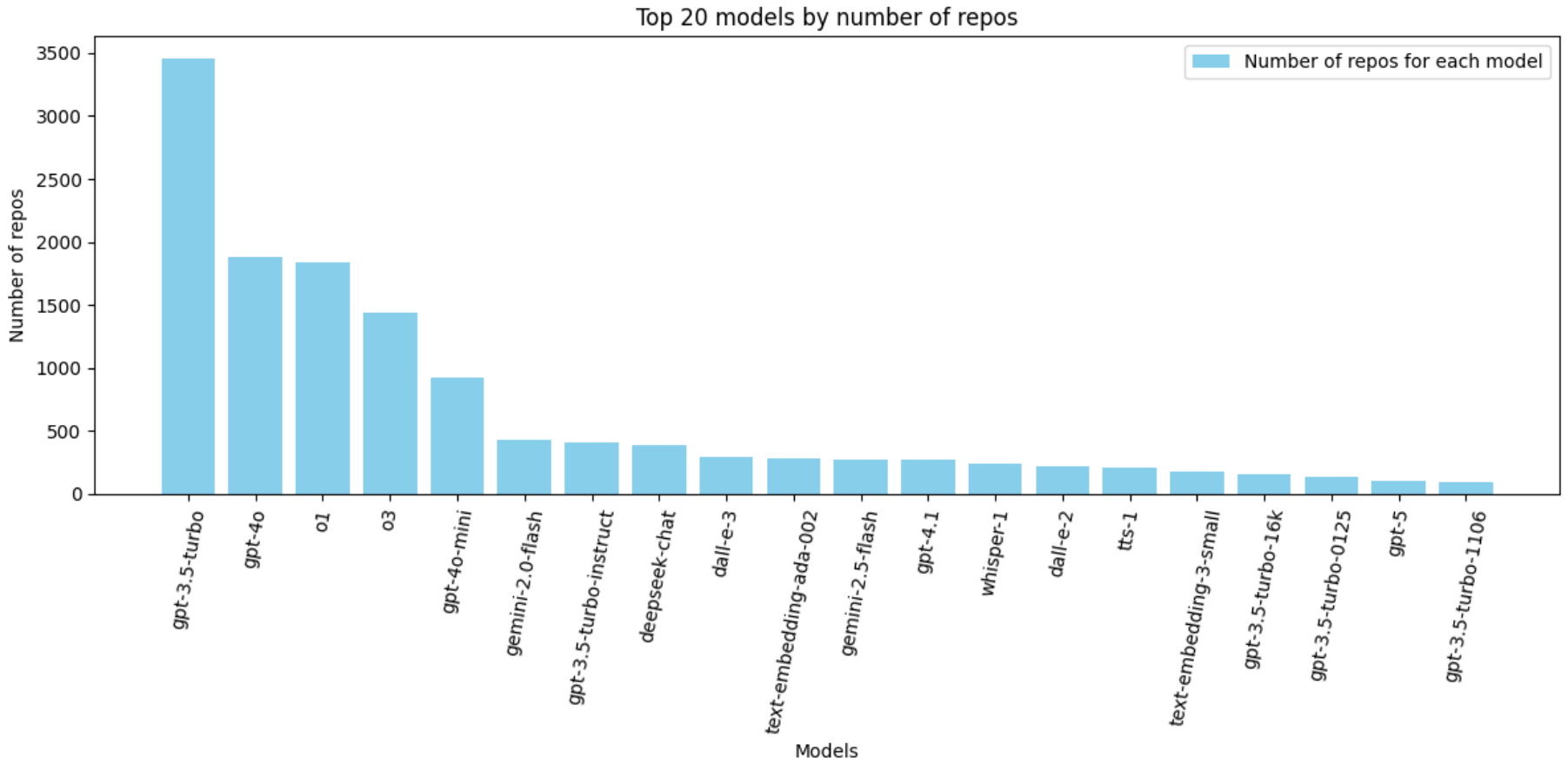
Library & Model Search – Go LLM SDK popularity



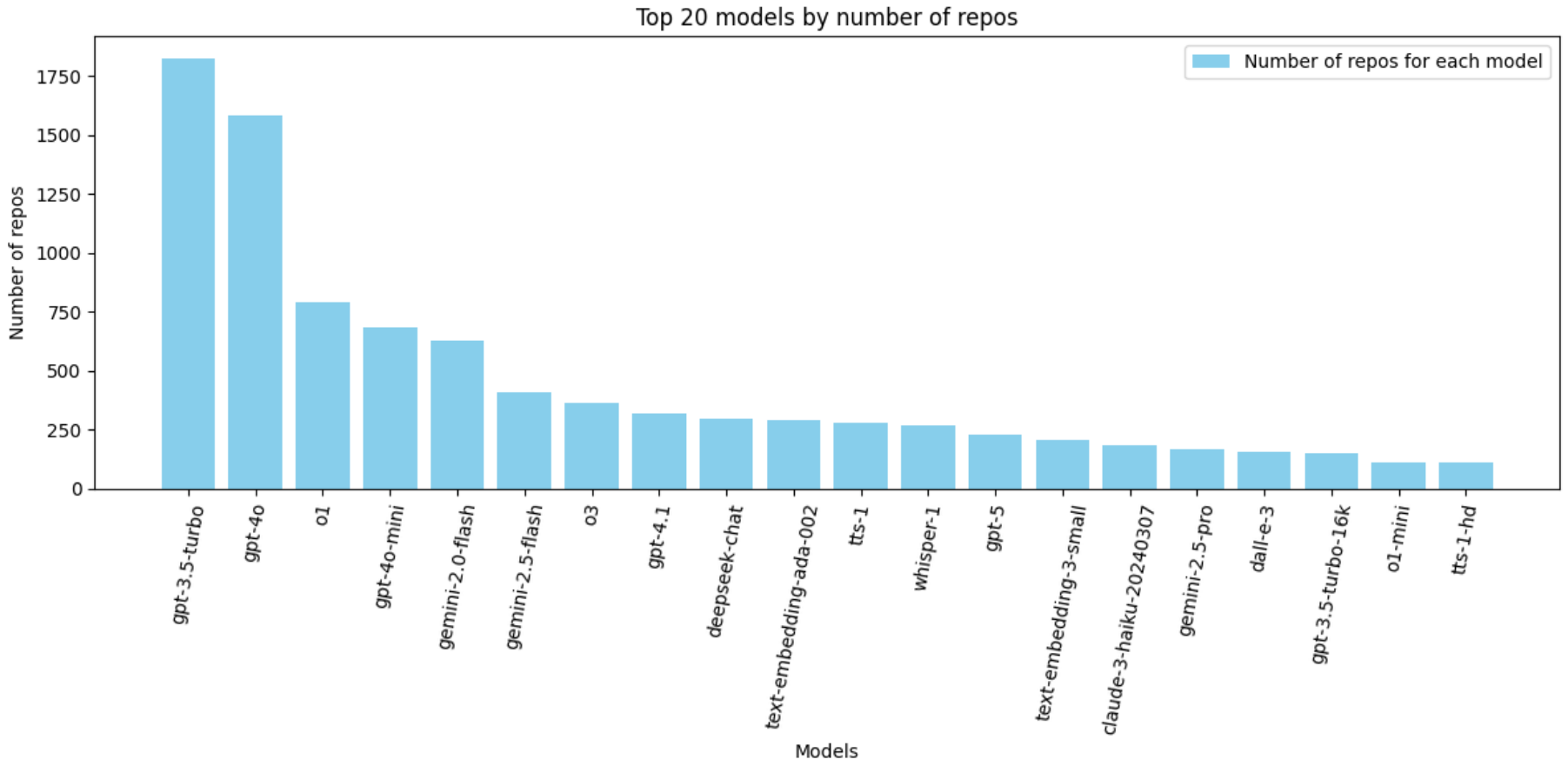
Library & Model Search – Python frequency of top models



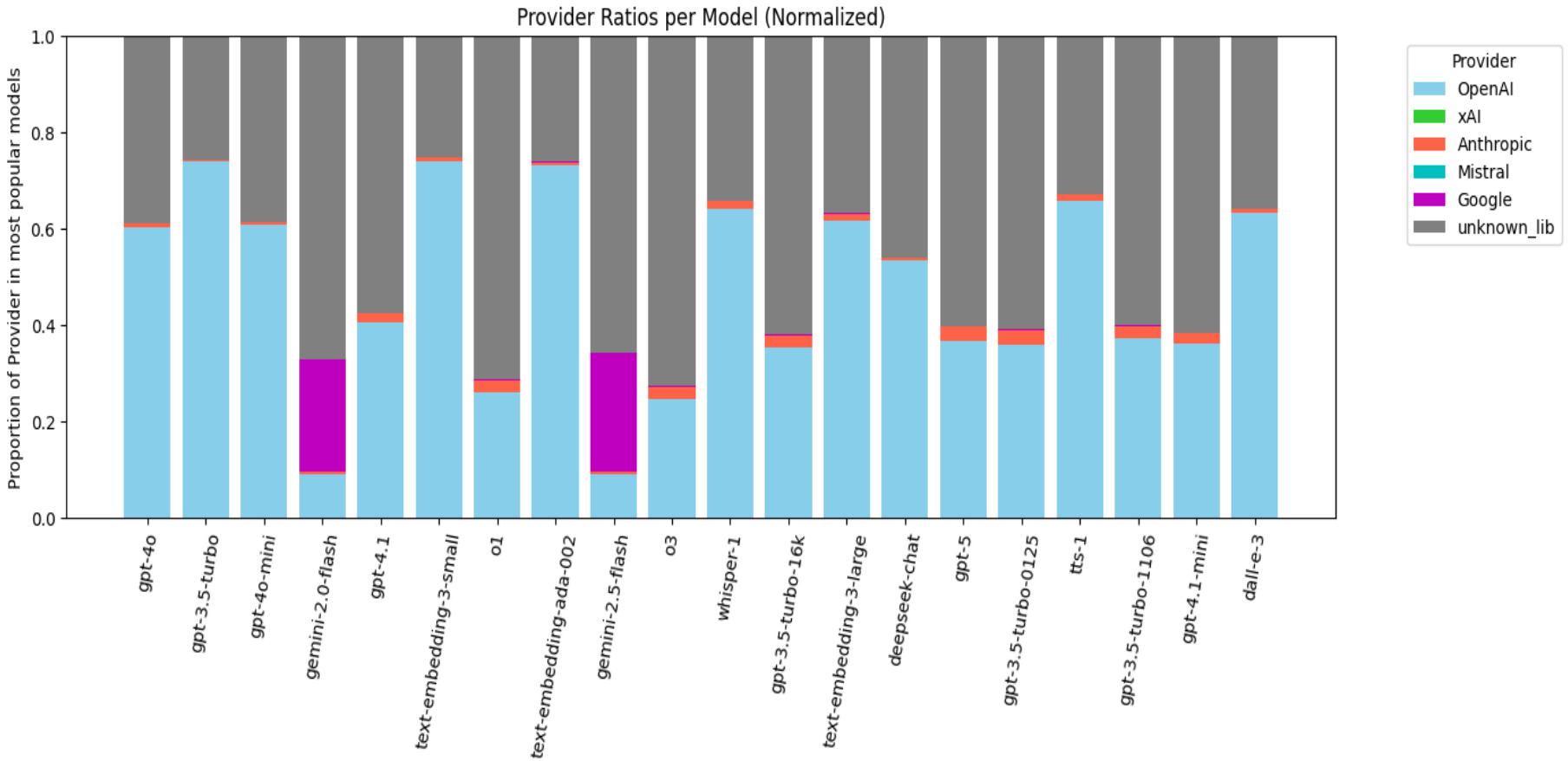
Library & Model Search – Java frequency of top models



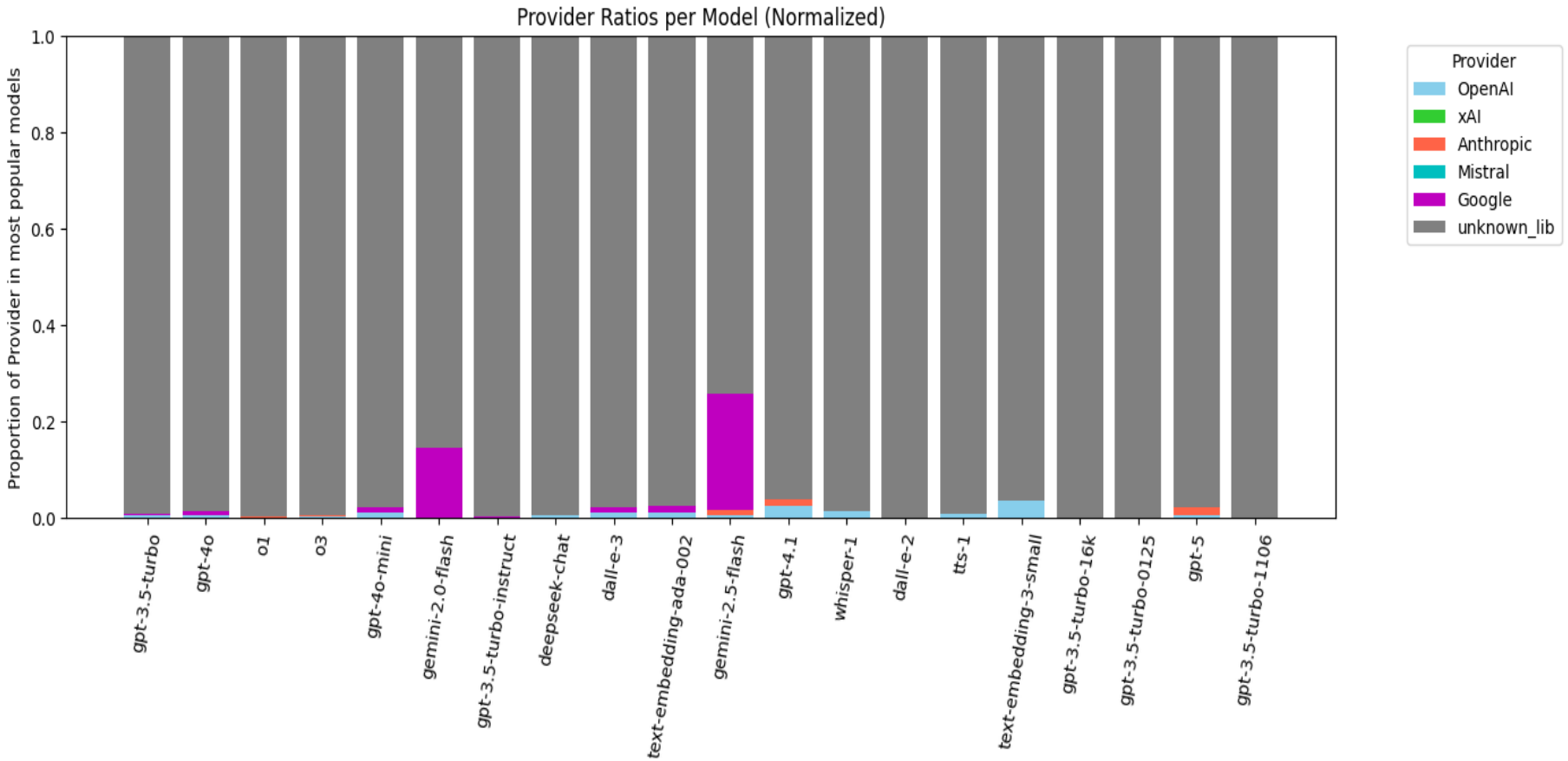
Library & Model Search – Go frequency of top models



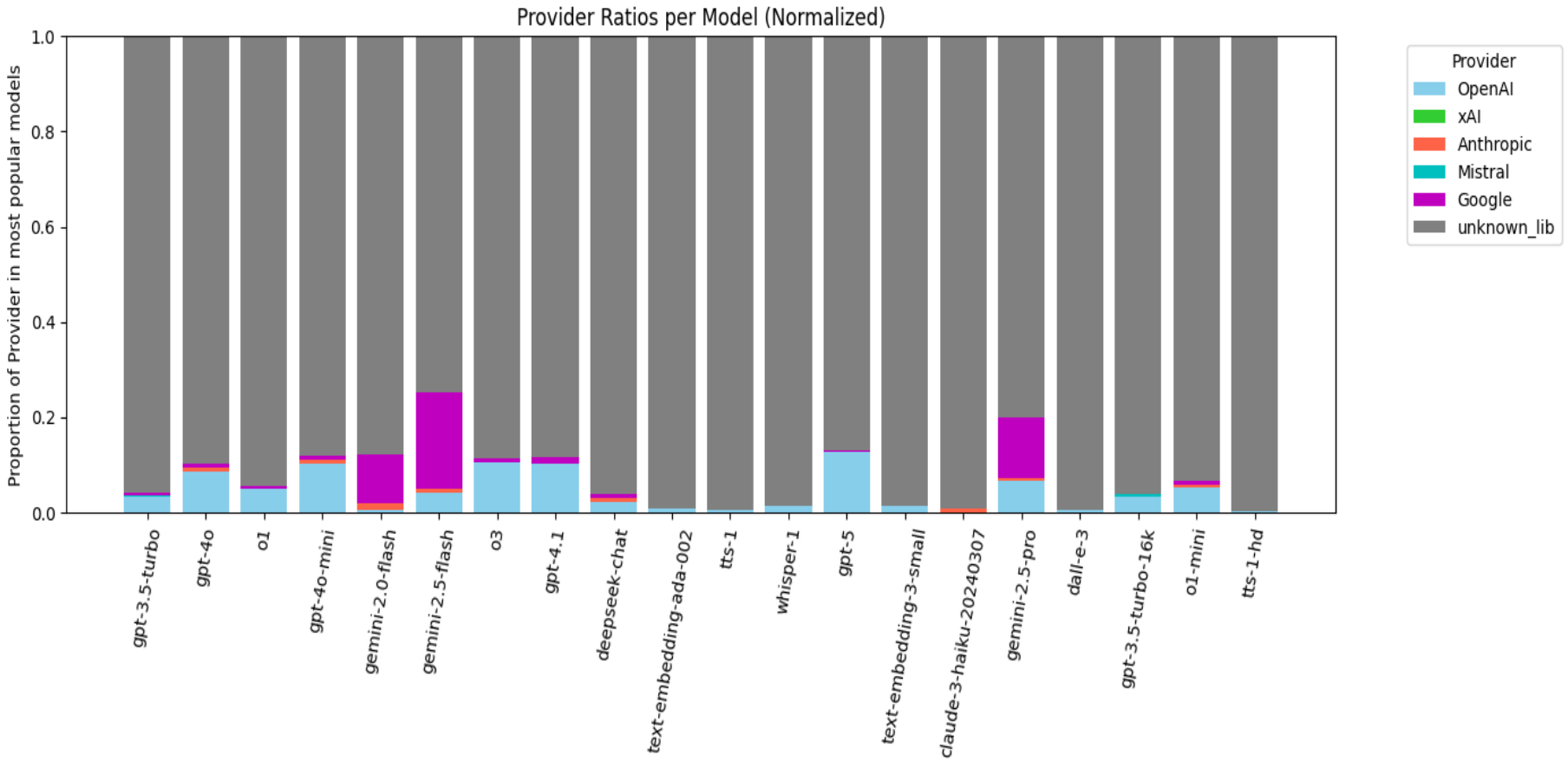
Library & Model Search – Python top LLM models with SDK provider



Library & Model Search – Java top LLM models with SDK provider



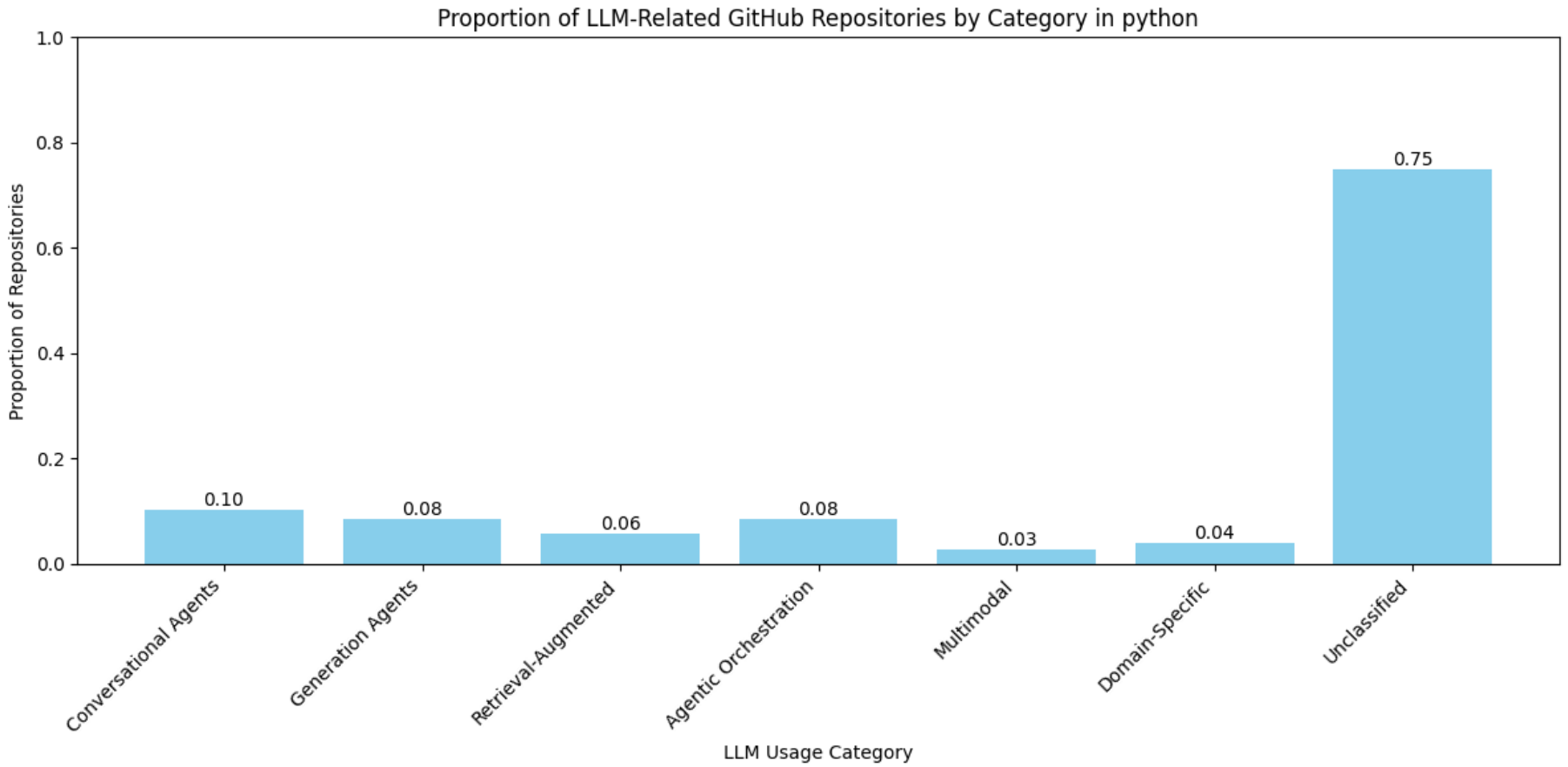
Library & Model Search – Go top LLM models with SDK provider



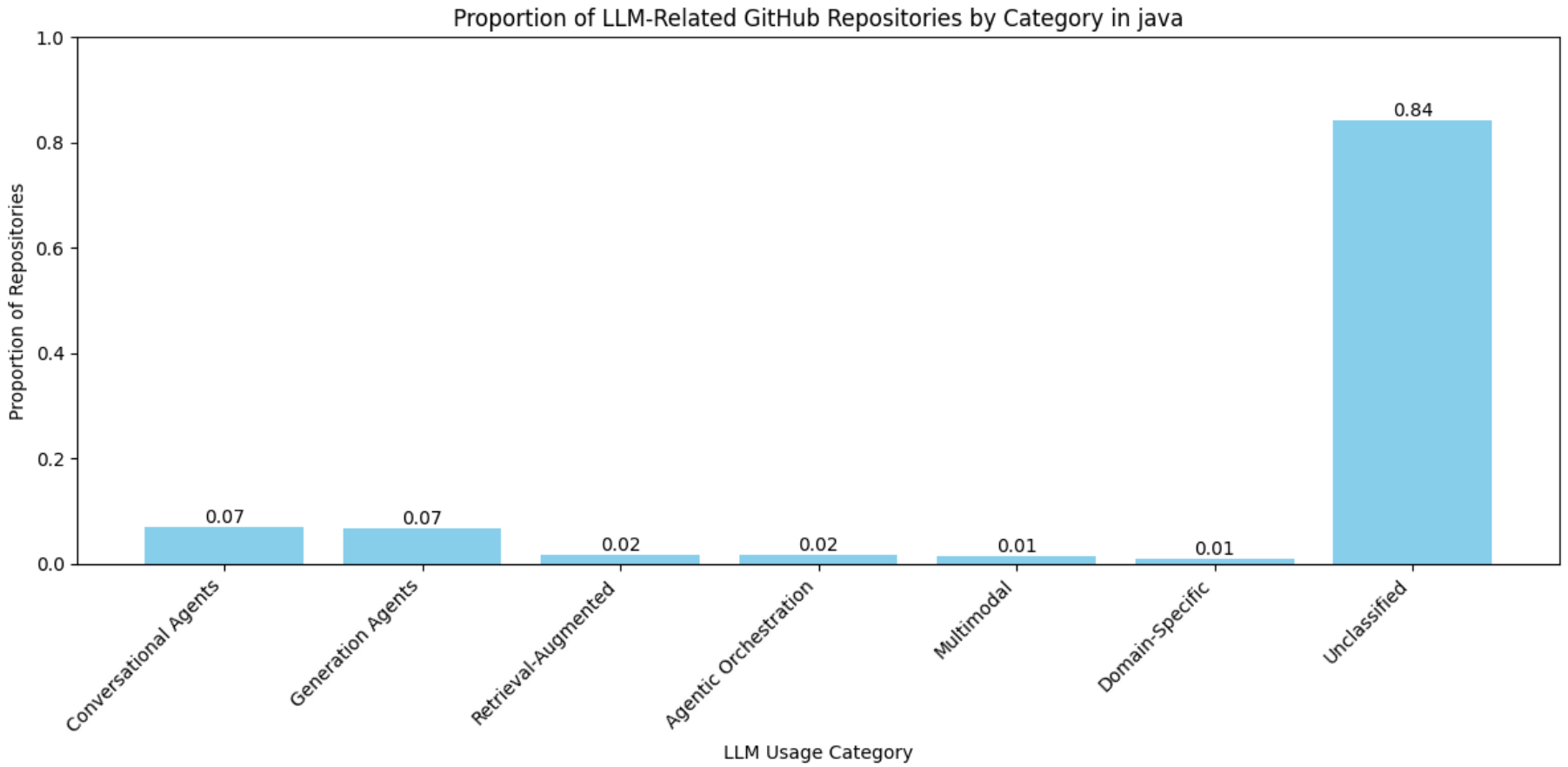
Attribute Search

We randomly sample 500 repository names from the previously collected repo population and query GitHub's repository endpoint to gather attributes and make statistical inferences on the whole population.

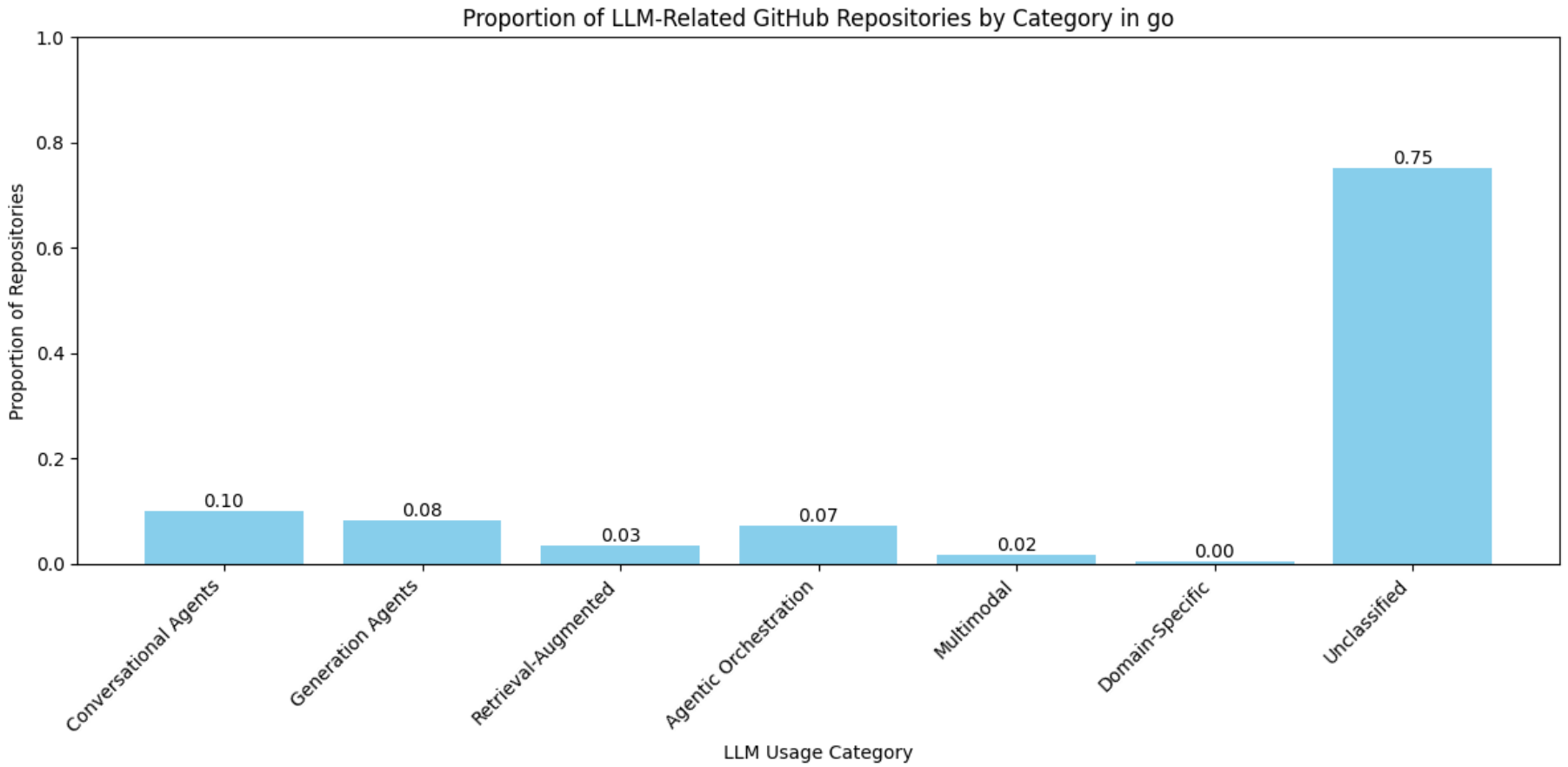
Attribute Search – Python LLM usage modes



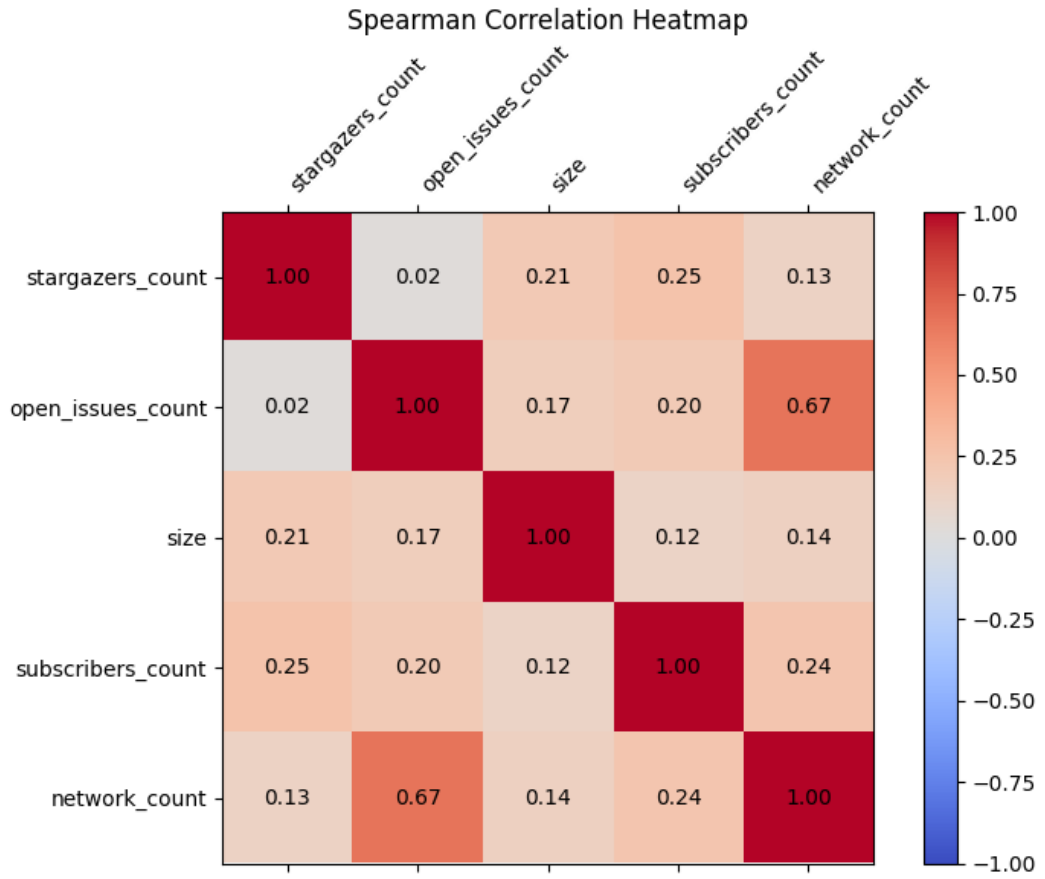
Attribute Search – Java LLM usage modes



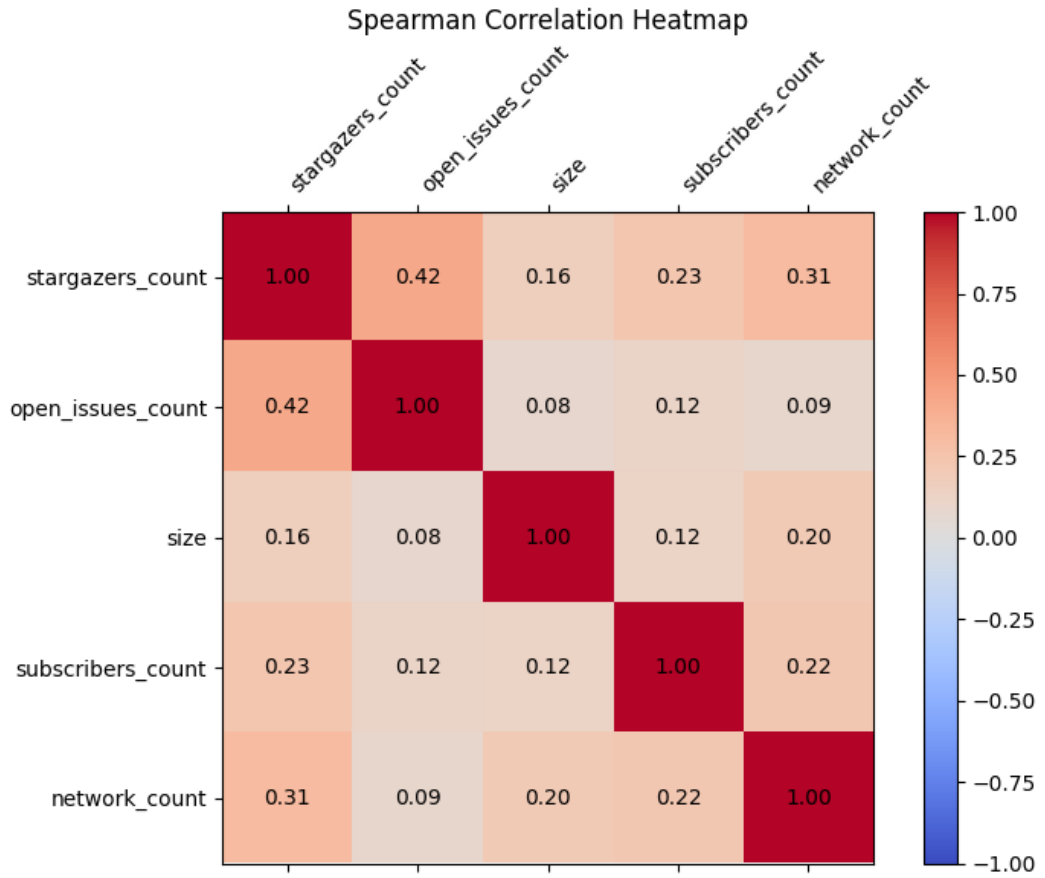
Attribute Search – Python LLM usage modes



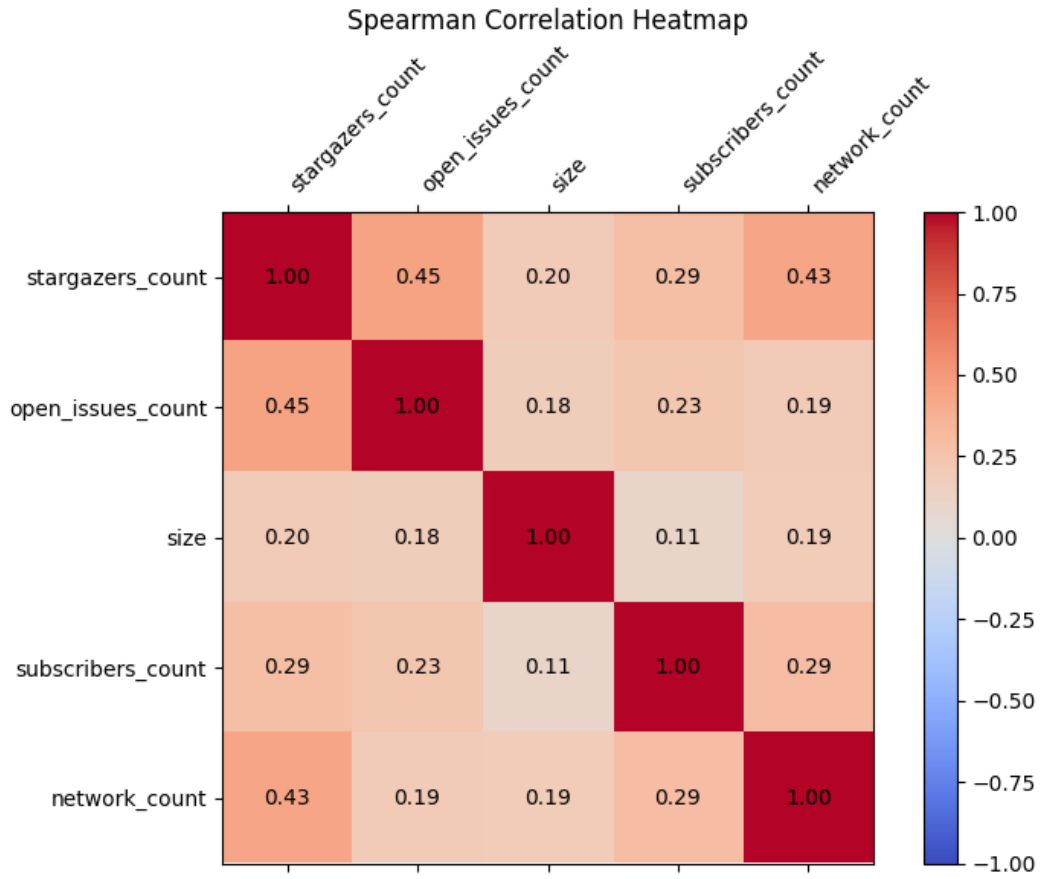
Attribute Search – Python Repositories Correlation of Attributes



Attribute Search – Java Repositories
Correlation of Attributes



Attribute Search – Go Repositories
Correlation of Attributes



First test (Python vs Java):

- H_0 — Null Hypothesis
  The mean number of stars in Python LLM Repositories is equal to the mean number of stars in Java LLM Repositories.
- H_1 — Alternative Hypothesis
  The mean number of stars in Python LLM Repositories is less than the mean number of stars in Java LLM Repositories.

Second test (Python vs Golang):

- H_0 — Null Hypothesis
  The mean number of stars in Python LLM Repositories is equal to the mean number of stars in Golang LLM Repositories.
- H_1 — Alternative Hypothesis
  The mean number of stars in Python LLM Repositories is less than the mean number of stars in Golang LLM Repositories.

Attribute Search – Statistical tests Star Number

We want to test whether Python LLM projects had a lower mean number of stars than projects from the other two languages.

First test (Python vs Java):

- H_0 — Null Hypothesis
  The mean number of stars in Python LLM Repositories is equal to the mean number of stars in Java LLM Repositories.
- H_1 — Alternative Hypothesis
  The mean number of stars in Python LLM Repositories is less than the mean number of stars in Java LLM Repositories.

Second test (Python vs Golang):

- H_0 — Null Hypothesis
  The mean number of stars in Python LLM Repositories is equal to the mean number of stars in Golang LLM Repositories.
- H_1 — Alternative Hypothesis
  The mean number of stars in Python LLM Repositories is less than the mean number of stars in Golang LLM Repositories.

Attribute Search – Statistical tests Star Number

Before starting the hypothesis testing procedure, we need to confirm whether the underlying distribution of the sample is normal ...

before Box-Cox

- $\overline{X}_{Python} = 18.99$
- $S^2_{Python} = 270.93^2$
- $\overline{X}_{Java} = 25.34$
- $S^2_{Java} = 179.75^2$
- $\overline{X}_{Go} = 188.41$
- $S^2_{Go} = 1986.66^2$

Sample statistics after Box-Cox:

- $\overline{X'}_{Python} = 0.173$
- $S'^2_{Python} = 0.268^2$
- $\overline{X'}_{Java} = 0.245$
- $S'^2_{Java} = 0.352^2$
- $\overline{X'}_{Go} = 0.393$
- $S'^2_{Go} = 0.506^2$

Lambda parameter for Box-Cox transformation:

- $\lambda_{Python} = -0.7517$
- $\lambda_{Java} = -0.6540$
- $\lambda_{Go} = -0.4958$

Attribute Search – Statistical tests Star Number

We applied the Anderson-Darling test and confirmed that the sample set was not normally distributed. We thus applied the Box-Cox transformation to normalize the sample data.

$$\alpha = 0.05$$

- Python vs Java —
  $t = -3.6139$
  $p = 0.000159$

- Python vs Go —
  $t = -8.5982$
  $p = 0.000000$

Attribute Search – Statistical tests Star Number

To compare the mean star numbers for each language, we applied Welch's t-test. The results confirmed our suspicion that Python LLM projects had a lower number of stars and thus were less popular than LLM projects in the other two languages.

size test (Python vs Java):

- H_0 — Null Hypothesis
  The mean size of Python LLM Repositories is equal to the mean size of Java LLM Repositories.
- H_1 — Alternative Hypothesis
  The mean size of Python LLM Repositories is less than the mean size of Java LLM Repositories.

Second size test (Python vs Golang):

- H_0 — Null Hypothesis
  The mean size of Python LLM Repositories is equal to the mean size of Golang LLM Repositories.
- H_1 — Alternative Hypothesis
  The mean size of Python LLM Repositories is less than the mean size of Golang LLM Repositories.

Attribute Search – Statistical tests Size

The statistical test for size comparison were conducted in the same way as the ones for star number.

size test (Python vs Java):

- H_0 — Null Hypothesis
  The mean size of Python LLM Repositories is equal to the mean size of Java LLM Repositories.
- H_1 — Alternative Hypothesis
  The mean size of Python LLM Repositories is less than the mean size of Java LLM Repositories.

Second size test (Python vs Golang):

- H_0 — Null Hypothesis
  The mean size of Python LLM Repositories is equal to the mean size of Golang LLM Repositories.
- H_1 — Alternative Hypothesis
  The mean size of Python LLM Repositories is less than the mean size of Golang LLM Repositories.

Attribute Search – Statistical tests Size

We first tested for normality using the Anderson-Darling test, and concluded that the sample was not normal. We then applied the Box-Cox transformation to obtain a more normal distribution.

$$\alpha = 0.05$$

- Python vs Java —
  $t = -5.5173$
  $p = 0.000000$
- Python vs Go —
  $t = -2.0314$
  $p = 0.021239$

Attribute Search – Statistical tests Size

The Welch t-test told us that Python projects were generally smaller in size than LLM projects in the other two languages. This reinforces the hypothesis that Python LLM projects are mostly experimental projects with activity driven by contributor engagement rather than the number of stars.

Conclusion

This study offers a comprehensive observational analysis of LLM adoption in public GitHub repositories, where Python dominates LLM-integrated development, with two orders of magnitude more repositories than Java or Go, while OpenAI SDK and models are the most widely adopted across all languages. The usage patterns are primarily Conversational, Generation, and Agentic Orchestration, reflecting common application Scenario.

We hope that this research will be usefull for: LLM-integrated software developers, to help them in selecting LLM models and SDKs with the largest community support, and for analyzing the current enviroment of LLM-integrate software in Github.