Final Project - Algorithms for Decision Making

Vidushi, Junran, Sat

10/28/2025

Introduction

GitHub has become one of the most important signals of programming language popularity. Developers star repositories to show interest, follow projects, and signal which languages they value. Understanding what factors drive GitHub popularity offers insight into how languages grow, why certain ecosystems thrive, and which features attract developer attention. In this project, we analyze the PLDB (Programming Language Database) TidyTuesday dataset, which includes over 4,000 programming languages and detailed metadata on GitHub activity, Wikipedia traffic, academic citations, estimated user counts, job market demand, and more.

Our primary question is: "Which features are most strongly associated with GitHub popularity, measured via GitHub repository stars?" To answer this, we construct and compare several statistical learning models—including regularized linear models (LASSO) and tree-based models (CART and Random Forests)—and evaluate their predictive performance and interpretability.

Our Dataset

The dataset includes 4,000+ programming languages from PLDB, with information on GitHub activity, Wikipedia traffic, academic references, estimated users, and job postings. Our outcome is GitHub popularity (star count). We selected ~20 predictors from these categories and excluded identifier fields (`pldb_id`, `title`). The variable `github_language_type` was removed due to >85% missing values and severe class imbalance.

Exploratory Data Analysis:

To understand the distribution of GitHub popularity and its relationship with other language characteristics, we explored summary statistics, univariate distributions, and several key bivariate relationships. Below are the primary patterns that informed our modeling choices.

Variable	Role	Type	Description
github_repo_stars	Response	Numeric	Number of GitHub stars on the language's main repo.
log10_github_stars	Response	Numeric	Log10-transformed GitHub star count used as the modeling response.
appeared	Explanatory	Numeric	Year the language was created or publicly announced.
language_rank	EDA Only	Numeric	Popularity rank from PLDB (lower = more popular).
github_repo_forks	Explanatory	Numeric	Number of forks of the main GitHub repository.
github_repo_subscribers	Explanatory	Numeric	Number of GitHub subscribers/watchers.
github_repo_issues	Explanatory	Numeric	Number of open issues in the GitHub repo.
github_language_repos	Explanatory	Numeric	Number of GitHub repositories using this language.

Variable	Role	Type	Description
wikipedia_daily_page_views	Explanatory	Numeric	Average daily Wikipedia page views.
wikipedia_backlinks_count	Explanatory	Numeric	Number of Wikipedia pages linking to this page.
wikipedia_revision_count	Explanatory	Numeric	Total number of Wikipedia edits.
book_count	Explanatory	Numeric	Number of books related to the language (ISBNdb).
semantic_scholar	Explanatory	Numeric	Count of academic papers referencing this language.
number_of_users	Explanatory	Numeric	Estimated user count (PLDB model).
number_of_jobs	Explanatory	Numeric	Estimated job postings requiring this language.
central_package_repository_count	Explanatory	Numeric	Number of packages in the central package repository.
is_open_source	Explanatory	Categorical	Whether the language is open source (TRUE/FALSE).
pldb_id	Identifier	Character	Standardized unique ID used by PLDB.
title	Identifier	Character	Official name of the programming language.

We treat `language_rank` as an exploratory variable only and exclude it from all predictive models, because it is too directly tied to GitHub popularity itself

```
##
##
## Table: Summary of Programming Language Variables
##
## |                                    | All Languages (N=889)  |     Total (N=889)      |
## |:-----------------------------------|:----------------------:|:----------------------:|
## |**github_repo_stars**               |                        |                        |
## |&nbsp;&nbsp;&nbsp;Mean (SD)         |  2127.403 (7554.016)   |  2127.403 (7554.016)   |
## |&nbsp;&nbsp;&nbsp;Range             |   0.000 - 88526.000    |   0.000 - 88526.000    |
## |**log10_github_stars**              |                        |                        |
## |&nbsp;&nbsp;&nbsp;Mean (SD)         |     2.261 (1.053)      |     2.261 (1.053)      |
## |&nbsp;&nbsp;&nbsp;Range             |     0.000 - 4.947      |     0.000 - 4.947      |
## |**appeared**                        |                        |                        |
## |&nbsp;&nbsp;&nbsp;Mean (SD)         |    2014.193 (6.225)    |    2014.193 (6.225)    |
## |&nbsp;&nbsp;&nbsp;Range             |  1981.000 - 2022.000   |  1981.000 - 2022.000   |
## |**github_repo_forks**               |                        |                        |
## |&nbsp;&nbsp;&nbsp;N-Miss            |           3            |           3            |
## |&nbsp;&nbsp;&nbsp;Mean (SD)         |   261.287 (1203.003)   |   261.287 (1203.003)   |
## |&nbsp;&nbsp;&nbsp;Range             |   0.000 - 23732.000    |   0.000 - 23732.000    |
## |**github_repo_subscribers**         |                        |                        |
## |&nbsp;&nbsp;&nbsp;N-Miss            |           4            |           4            |
## |&nbsp;&nbsp;&nbsp;Mean (SD)         |    62.342 (200.882)    |    62.342 (200.882)    |
## |&nbsp;&nbsp;&nbsp;Range             |    0.000 - 2910.000    |    0.000 - 2910.000    |
## |**github_repo_issues**              |                        |                        |
## |&nbsp;&nbsp;&nbsp;N-Miss            |          104           |          104           |
## |&nbsp;&nbsp;&nbsp;Mean (SD)         |   123.034 (546.255)    |   123.034 (546.255)    |
## |&nbsp;&nbsp;&nbsp;Range             |    0.000 - 9522.000    |    0.000 - 9522.000    |
## |**github_language_repos**           |                        |                        |
## |&nbsp;&nbsp;&nbsp;N-Miss            |          752           |          752           |
## |&nbsp;&nbsp;&nbsp;Mean (SD)         | 75699.745 (423425.114) | 75699.745 (423425.114) |
## |&nbsp;&nbsp;&nbsp;Range             |  0.000 - 3479326.000   |  0.000 - 3479326.000   |
## |**wikipedia_daily_page_views**      |                        |                        |
## |&nbsp;&nbsp;&nbsp;N-Miss            |          796           |          796           |
## |&nbsp;&nbsp;&nbsp;Mean (SD)         |   172.860 (441.441)    |   172.860 (441.441)    |
## |&nbsp;&nbsp;&nbsp;Range              |   -1.000 - 3151.000    |   -1.000 - 3151.000    |
## |**wikipedia_backlinks_count**        |                        |                        |
## |&nbsp;&nbsp;&nbsp;N-Miss             |          800           |          800           |
## |&nbsp;&nbsp;&nbsp;Mean (SD)          |   192.449 (840.854)    |   192.449 (840.854)    |
## |&nbsp;&nbsp;&nbsp;Range              |    2.000 - 7839.000    |    2.000 - 7839.000    |
## |**wikipedia_revision_count**         |                        |                        |
## |&nbsp;&nbsp;&nbsp;N-Miss             |          809           |          809           |
## |&nbsp;&nbsp;&nbsp;Mean (SD)          |   319.975 (1158.337)   |   319.975 (1158.337)   |
## |&nbsp;&nbsp;&nbsp;Range              |   1.000 - 10104.000    |   1.000 - 10104.000    |
## |**book_count**                       |                        |                        |
## |&nbsp;&nbsp;&nbsp;Mean (SD)          |     1.278 (11.845)     |     1.278 (11.845)     |
## |&nbsp;&nbsp;&nbsp;Range              |    0.000 - 274.000     |    0.000 - 274.000     |
## |**semantic_scholar**                 |                        |                        |
## |&nbsp;&nbsp;&nbsp;N-Miss             |          744           |          744           |
## |&nbsp;&nbsp;&nbsp;Mean (SD)          |     2.876 (6.540)      |     2.876 (6.540)      |
## |&nbsp;&nbsp;&nbsp;Range              |     0.000 - 36.000     |     0.000 - 36.000     |
## |**number_of_users**                  |                        |                        |
## |&nbsp;&nbsp;&nbsp;Mean (SD)          |  6966.435 (82892.106)  |  6966.435 (82892.106)  |
## |&nbsp;&nbsp;&nbsp;Range              |  1.000 - 2356101.000   |  1.000 - 2356101.000   |
## |**number_of_jobs**                   |                        |                        |
## |&nbsp;&nbsp;&nbsp;Mean (SD)          |   71.202 (1153.029)    |   71.202 (1153.029)    |
## |&nbsp;&nbsp;&nbsp;Range              |   0.000 - 30349.000    |   0.000 - 30349.000    |
## |**central_package_repository_count** |                        |                        |
## |&nbsp;&nbsp;&nbsp;N-Miss             |          646           |          646           |
## |&nbsp;&nbsp;&nbsp;Mean (SD)          |     0.000 (0.000)      |     0.000 (0.000)      |
## |&nbsp;&nbsp;&nbsp;Range              |     0.000 - 0.000      |     0.000 - 0.000      |
## |**is_open_source**                   |                        |                        |
## |&nbsp;&nbsp;&nbsp;N-Miss             |          643           |          643           |
## |&nbsp;&nbsp;&nbsp;FALSE              |        0 (0.0%)        |        0 (0.0%)        |
## |&nbsp;&nbsp;&nbsp;TRUE               |      246 (100.0%)      |      246 (100.0%)      |
```

After reviewing the table, we see that most variables are numerical and measured as counts (GitHub activity, Wikipedia views) or rankings (popularity rank). A few variables are categorical, including open-source status. Other potentially categorical variables (such as `github_language_type` and `origin_community`) were removed because they contained manymissing values and extremely unbalanced categories.

Model building:

We fit three models: LASSO, CART, and Random Forest, to evaluate both linear and nonlinear relationships and compare predictive accuracy. All tuning was performed using cross-validation.

Across the penalty grid, the cross-validated RMSE values were all around 0.91 on the log10 scale, with only small differences between penalties. Using RMSE as our optimization metric, the best penalty selected by cross-validation was approximately lambda = 0.002, which balances model complexity and shrinkage.

On the test set, our LASSO model obtained an RMSE of about 4.10 and an R^2 of about 0.02. Because the response is on the log10 scale, an RMSE this large indicates that the linear model is not capturing the nonlinear structure in the data. An R^2 of only about 2% means that this LASSO specification explains very little of the variation in log10 GitHub stars, especially when compared with the tree-based models we fit later.

The LASSO variable-importance plot shows that GitHub stars are most strongly associated with GitHub ecosystem indicators such as subscribers, forks, and number of repositories, while other predictors contribute relatively little.

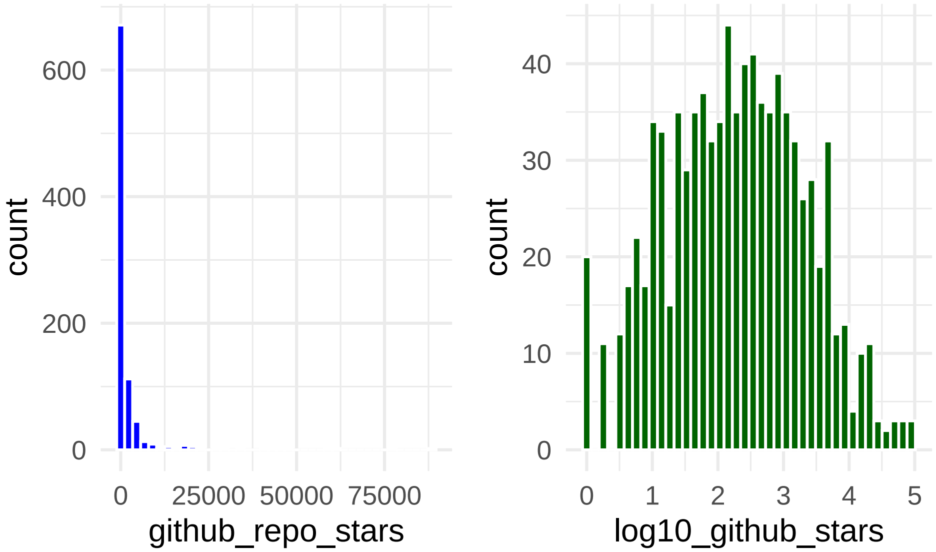


Figure 1: The raw GitHub star counts are highly right-skewed, while the log10 transformation produces a more symmetric distribution that is easier to model.

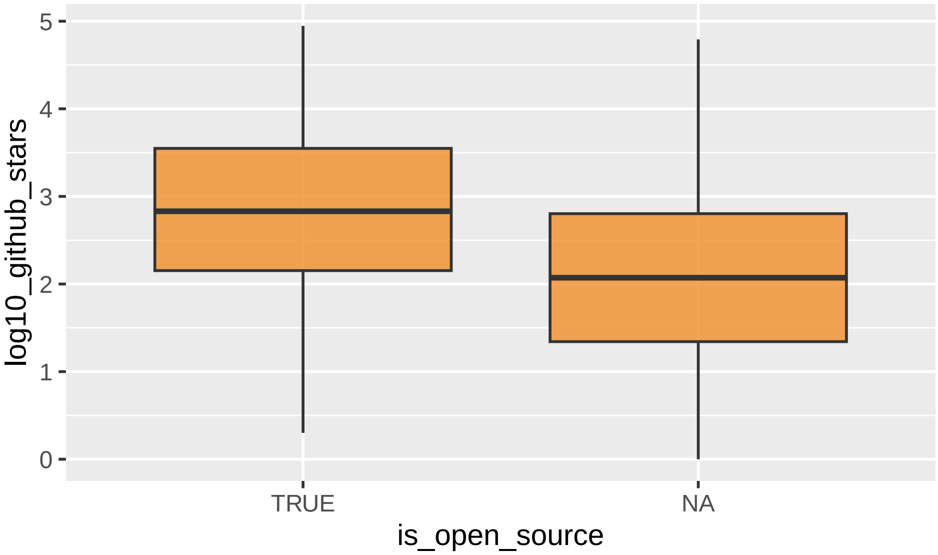


Figure 2: Open-source languages tend to have slightly higher star counts, but the large overlap shows that this variable alone is not strongly predictive.

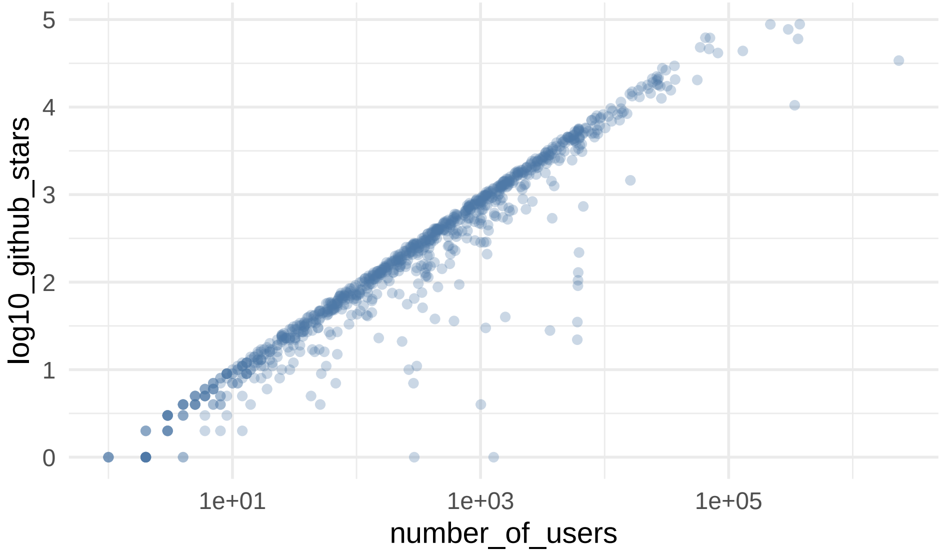


Figure 3: On a log scale, the number of users increases roughly linearly with GitHub stars, suggesting that community size is closely tied to popularity.

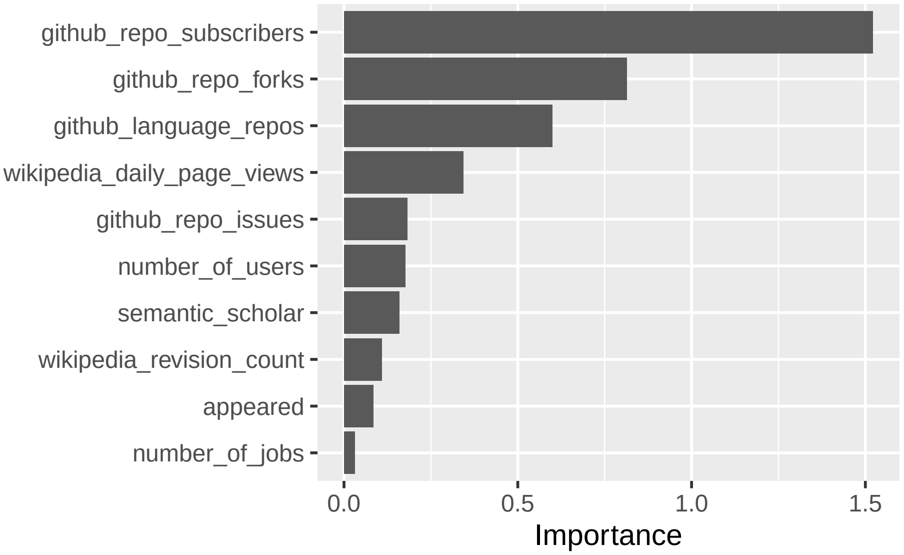


Figure 4: The LASSO model identifies GitHub ecosystem features; subscribers, forks, and repository counts, as the strongest predictors of star counts.

The LASSO model keeps several predictors. github__repo__subscribers = 1.48 gives the strongest positive effect, meaning languages that attract more subscribers tend to receive more GitHub stars. wikipedia__daily__page__views = 0.318 adds a smaller positive contribution, so higher public visibility relates to more stars. github__repo__issues = 0.158 adds a modest positive effect, while book__count = 0.0077 contributes almost none. The model assigns the strongest negative effect to github__repo__forks = -0.772, so once other visibility signals are accounted for, higher fork counts predict fewer stars. github__language__repos = -0.566, semantic__scholar = -0.150, number__of__users = -0.146, appeared = -0.086, and smaller negative coefficients reduce predicted stars slightly. number__of__jobs shrinks to zero.

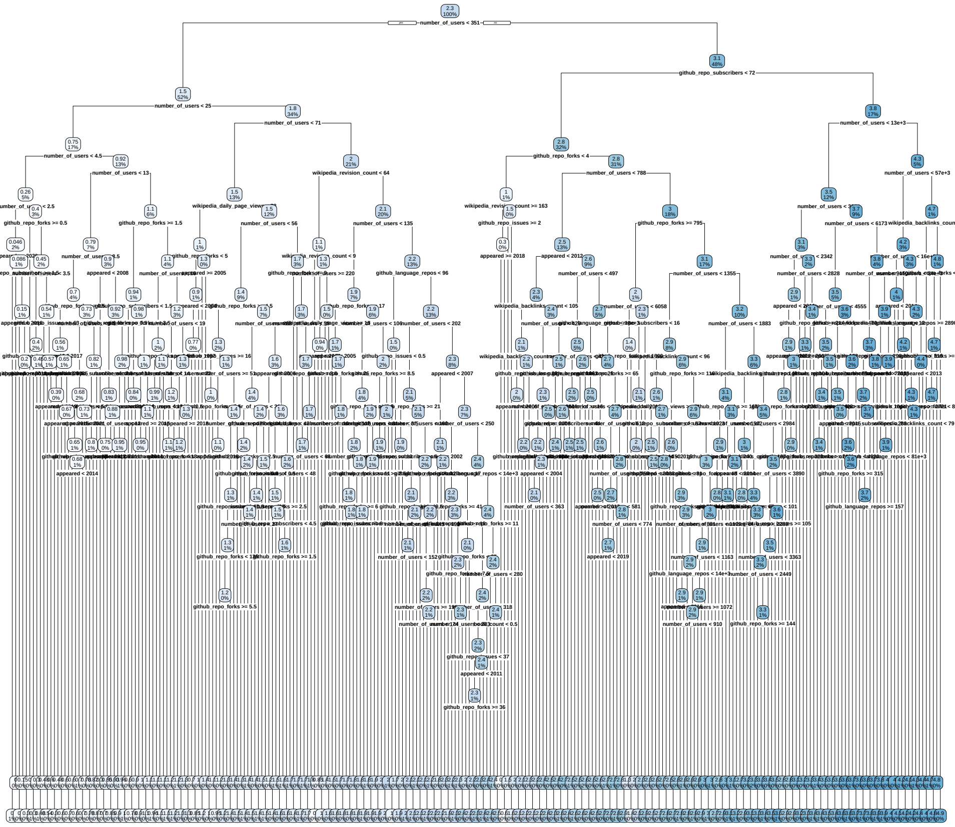


Figure 5: CART decision tree for predicting log10 GitHub stars.

The CART model achieved a high Rˆ2 (around 0.94) with similar cross-validated and test performance, suggesting that the predictors contain strong signal rather than noise.

Random Forest

The tuned random forest achieved an RMSE of 0.142 and an R^2 of 0.982 on the test set, substantially outperforming the LASSO model. Its variable-importance plot highlights number__of__users, github__repo__subscribers, github__repo__forks, and github__repo__issues as the strongest predictors, reinforcing the role of community size and GitHub activity.

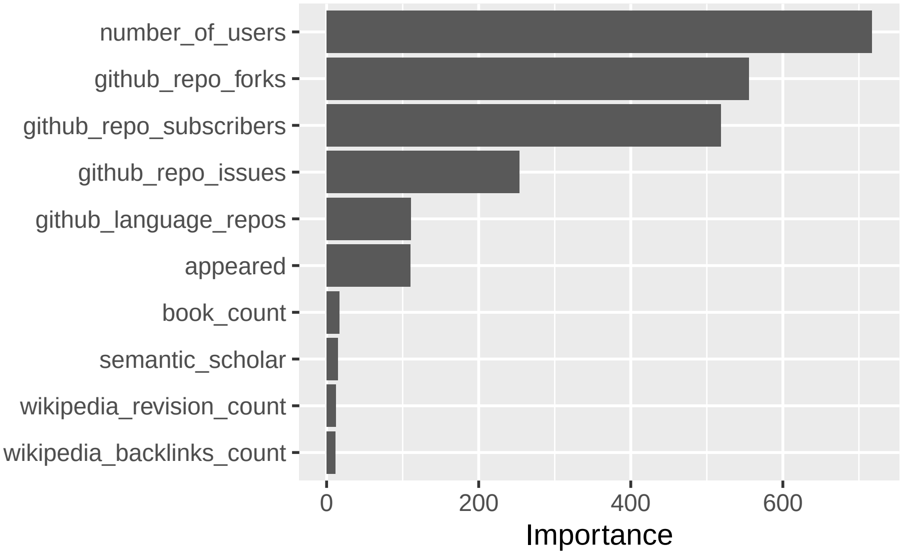


Figure 6: The CART model also highlights forks, subscribers, and number of users as the most important predictors, consistent with the LASSO results.

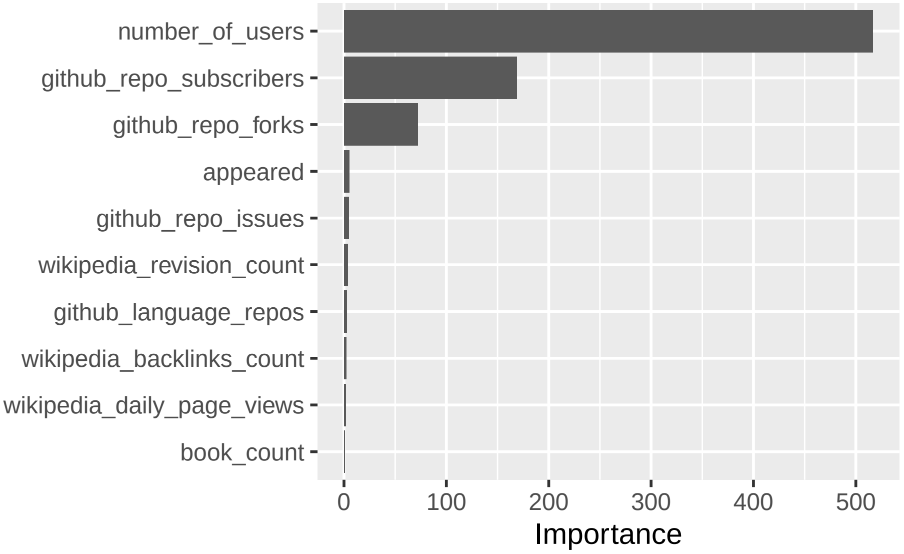


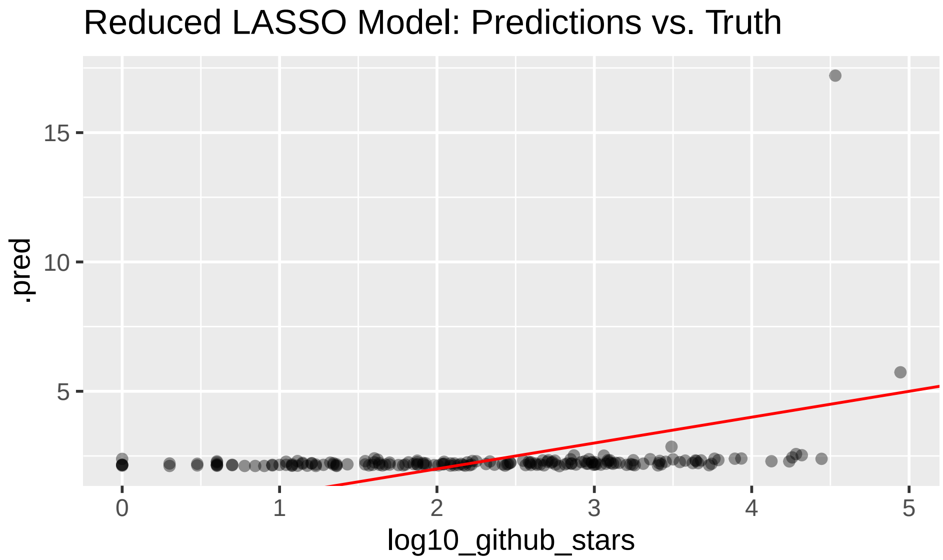
Figure 7: The random forest confirms the dominant role of community size and GitHub activity indicators, with number of users as the most influential variable.

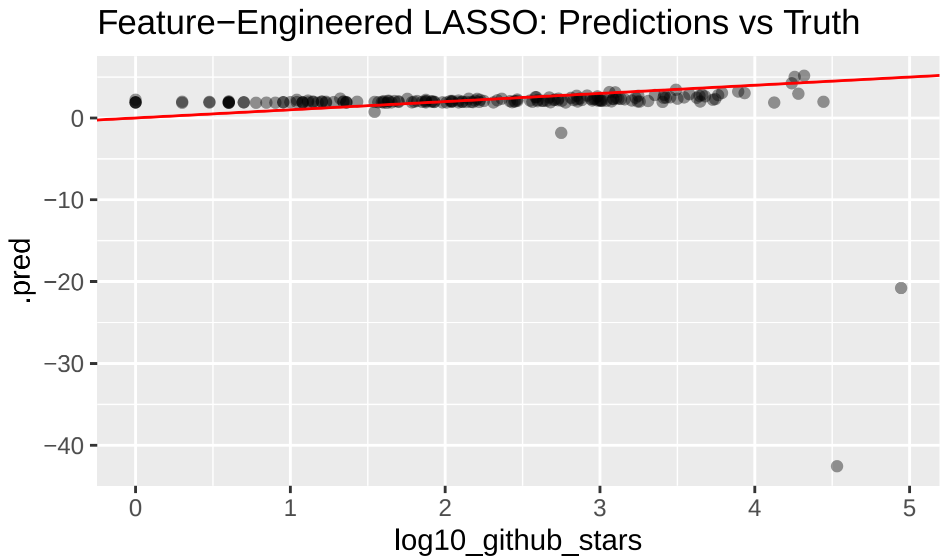
Model refinement:

- Across our LASSO and CART models, the most important variables are github_repo_subscribers, github_repo_forks, github_language_repos, number_of_users.

To see how much predictive power comes specifically from core GitHub variables, we refit a reduced LASSO model that removed github_repo_subscribers, github_repo_forks, github_language_repos, and wikipedia_daily_page_views from the predictors. We had already excluded language_rank from all models, since it is essentially a popularity summary too closely tied to GitHub stars themselves.

To see if we could improve the model, we created a new variable years_since_release = 2025 - appeared, which measures how long a language has been around. However, the test RMSE (about 4.10) and R^2 (about 0.02) remained essentially unchanged, suggesting that age adds little beyond what the existing ecosystem variables already capture.





Because the removed variables encode core GitHub ecosystem activity, the reduced model is forced to rely on weaker predictors and cannot accurately distinguish highly popular languages from obscure ones.

Conclusion:

Table 2: Summary of model performance on the test set.

Model	RMSE	R2	Top_Predictors
LASSO	4.10	0.02	subscribers, forks, repos, wiki_views
CART	0.22	0.96	forks, subscribers, number_of_users
Random Forest	0.14	0.98	number_of_users, subscribers, forks, issues
Reduced LASSO	1.37	0.06	weak predictors only
Feature-Engineered LASSO	4.10	0.02	same as LASSO + years_since_release

Our analysis shows that GitHub popularity is driven overwhelmingly by indicators of community size and ecosystem activity, rather than by metadata such as age, job postings, books, or academic citations. Across models, the same predictors consistently dominate: number_of_users, github_repo_subscribers, github_repo_forks, github_language_repos, and wikipedia_daily_page_views. Tree-based models best capture the nonlinear patterns needed for strong predictive accuracy, while more interpretable linear approaches like LASSO underperform. Feature engineering adds little once core GitHub ecosystem signals are removed, though future work with time-series data, boosted tree methods, or network-based features could further improve explanatory power. Overall, our results suggest that developer community size and GitHub activity are the strongest determinants of programming language popularity.