What Predicts Model Adoption? Browse-Time Signals and AI Model Selection on Hugging Face

FEM11152 – Seminar Data Science for Marketing Analytics

Maggie Tsou

Jan, 2026

1. Introduction

Open-source platforms democratize innovation by allowing anyone to share their work and letting contributions to rise based on merit. Hugging Face represents one of the most well-known example of this concept in the AI industry with the largest repository of pre-trained models being developed by researchers, companies and individuals globally. By eliminating traditional barriers, this platform made it possible for users to share a model that reaches millions of developers and organizations within minutes. The transformation of this diffusion is closely connected with AI model development, specifically, the efficiency of it, as developers can select a pre-trained model (transfer learning) as opposed to training a new model from scratch. However, democratized access to a large library of models does not translate to immediate success. Since AI models vary enormously in quality, computational requirements, and suitability for different tasks, when developers face 500,000 model options, information asymmetry becomes severe due to the fact that developers cannot evaluate every alternative. As Parker et al. (2016) observed, while platforms reduce distribution costs, they do not eliminate discovery costs. Developers therefore rely on observable signals at browse-time such as creator details, model documentation, and technical specifications to guide selection decisions. This study explores the following research question: To what extent can AI model adoption be predicted from browse-time data, and which observable features most strongly predict adoption outcomes on large open-source platforms? Predictive modeling with Naïve Bayes as a baseline classifier and XGBoost for maximum accuracy are used to perform this analysis. SHAP (SHapley Additive exPlanations) values are then used to examine feature contributions from XGBoost. Understanding both the data features that best predicts adoption as well as the features that most contribute has implications for model contributors seeking visibility, enterprises evaluating innovation options, and platforms like Hugging Face in designing platform algorithms.

2. Data

Data used in this study were obtained from a publicly available Kaggle dataset of Hugging Face AI models, containing a random sample of models from the Hugging Face platform. After excluding models with missing download or temporal data, the final analytical sample consists of 45,707 models with five variables: URL, title, downloads, likes, and last update timestamp.

First, following established research practices in open-source innovation where usage metrics are commonly served as indicators of adoption decisions (Lerner & Tirole, 2002; Dabbish et al., 2012), is_high_adoption is implemented as a binary outcome. Download counts in string formats (e.g., “K”,

"M") are parsed into numeric values, and models with cumulative downloads in the top 25% of the distribution are classified as high adoption models.

Then, feature engineering extracted organizational, technical, and temporal predictors from URL, title, and last update timestamp. Organizational characteristics include *n_models_by_author*, which counts the total contributions by each creator, and *author_type*, classified into big_tech (Google, Meta, Microsoft, OpenAI, NVIDIA, etc; 3.2%), ai_company (specialized AI firms; 0.3%), organization (universities, research institutes; 13.4%), and individual developers (83%). Technical characteristics were extracted from model titles using keyword matching: *model_family* (BERT 20.3%, GPT 6.1%, T5 7.8%, RoBERTa 6.9%, Other 51.2%), *model_size* (tiny through xxl), *is_finetuned*, *is_distilled*, *domain*, and *language*. Temporal feature include *model_age_days*, converted from the timestamp and measuring since the model was last updated. Feature engineering actions yielded 10 total variables, with *is_high_adoption* serving as the outcome variable. The original variables (URL, title, and last update timestamp) were subsequently excluded as the relevant content had been extracted, and the variable *likes* were excluded as they reflect popularity that accumulates alongside downloads but not the characteristics observable at the time of model selection. As such, *likes* would capture contemporaneous feedback effects rather than the underlying features that drive adoption decisions.

During preprocessing, the continuous variable, *n_models_by_author*, showed high skewness and was log-transformed to satisfy assumptions of the baseline model, while *model_age_days* was retained in its original scale since the distribution is bimodal and correctly reflects the platform's dynamics. Categorical variables were encoded as factors, then these factor levels were synchronized between training and test sets for consistent feature representations during model training and evaluation. All preprocessing steps relied only on information observable at the platform level and does not incorporate post-adoption or external performance data.

3. Methods

Two complementary machine learning methods are used in this study for the classification prediction of adoption. Naïve Bayes is a classic algorithm for classification tasks. It estimates the likelihoods of individual predictors ($P(x_j \mid Y=k)$ then combine them to make the final prediction, though operating under the conditional assumption that variables are independent of one another, making it "naïve".

$$\hat{y} = \arg \max_{k \in \{1,\ldots,K\}} \left[\log P(Y = k) + \sum_{j=1}^{p} \log P(x_j \mid Y = k) \right].$$

This classifier is employed as baseline due to its simplicity and efficiency for later comparison with a more flexible and complex model, and if Naïve Bayes performs comparably with it, adoption signals are likely to be independent. Whereas, a substantial performance improvement from the more complex model would indiciate that interactions amongst the features may be important drivers of adoption, making Naïve Bayes an adequate benchmark.

XGBoost (Gradient Boosting) is the more intricate model chosen for this analysis. The algorithm sequentially builds many weak decision trees, where at each iteration t, a new tree $f_t(\mathbf{x})$ is added to the existing prediction $\hat{y}^{(t-1)}$ to reduce overall loss. Each new tree corrects the error made by the previous one then combined, maximizing predictive accuracy. In contrast to Naïve Bayes, it is able to naturally handle feature interactions because trees split on combinations of features and therefore is able to capture complex, non-linear patterns. It also accomodates for this dataset's class imbalance and provide built-in regularization $\Omega(f_t)$ to prevent overfitting.

$$\mathcal{L} = \sum_{i=1}^{n} \ell\left(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)\right) + \Omega(f_t), \quad \Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2.$$

As XGBoost is more difficult to interpret directly, SHAP (SHapley Additive exPlanations) values is employed for the purpose of decomposing the predictions to quantify each feature's marginal contribution. SHAP assigns importance values based on cooperative game theory, and mean absolute SHAP values aggregate individual predictions to produce global feature importance rankings, revealing the factors that developers rely on most heavily when making selection decisions.

4. Analyses & Results

4.1 Naïve Bayes

Gaussian Naive Bayes requires continuous features to approximate normality, so log-transformation is performed on the author productivity variable (*n_models_by_author*) that exhibits severe right skew (skewness = 2.59). Categorical predictors are converted as factors with Laplace smoothing (α = 1) applied to prevent zero-probability estimates for rare category combinations. Classification analysis requires converting predicted probabilities to discrete labels with a decision threshold, of which is optimized by evaluating F1 scores, chosen as it is a threshold-dependent metric that balances precision and recall under class imbalance. across candidates from 0.10 to 0.90. The optimal threshold for Naive Bayes is 0.19, reflecting the model's tendency toward overconfident predictions of the majority class.

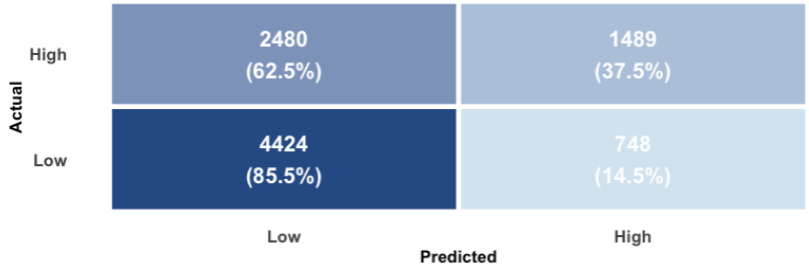

Figure 2: Naïve Bayes Confusion Matrix

The model's performance shows an AUC of 0.71, meaning that the model ranks a randomly selected high adoption model above a randomly selected low adoption model approximately 71% of the time. However, classification performance is poor. Precision of 0.38 means that few models predicted as high adoption

actually achieve it, while recall of 0.67 showing one-third of successful models are missed entirely. The F1 score of 0.48 clearly reflects this imbalance between false positives and false negatives. (Figure 1) The model is overconfident because the independence assumption is violated, as Chi-square tests confirm (Figure 2), features are strongly dependent, which inflated predictions toward whichever class those features favor. This violation of assumption motivates the usage of methods that accommodate feature interactions.

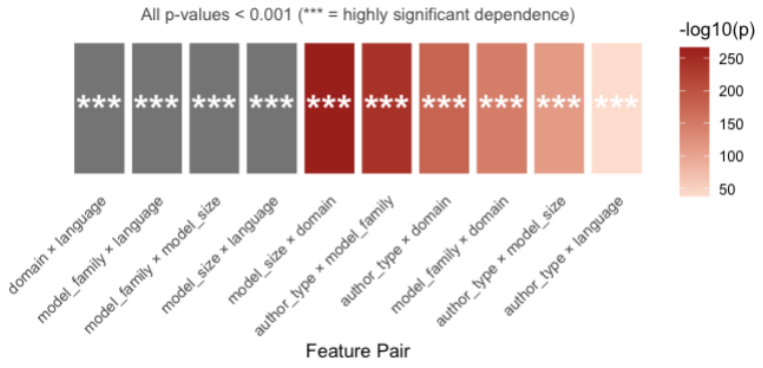

Figure 2: Chi-Square Independence Test on Naïve Bayes

4.2 XGBoost

XGBoost requires numeric input matrices, requiring one-hot encoding of categorical variables. Class imbalance is addressed through the scale_pos_weight parameter, set to the ratio of negative to positive instances (approximately 3.1), which gives more weight to the minority class. Hyperparameters are tuned using sequential optimization with 5-fold cross-validation, a tuning strategy that reduces computational heaviness of a full grid search. The tuning process is performed in four stages:

Stage 1: Maximum tree depth (4, 6, 8, 10, 12) is evaluated with learning rate fixed at 0.1. Optimal depth = 8 is identified after cross-validation AUC

Stage 2: Evaluate learning rates of 0.01, 0.03, 0.05, 0.1, 0.2. Smaller rates require more boosting iterations but often achieve better final performance. The value $\eta = 0.05$ maximizes cross-validation AUC, with early stopping determining the number of rounds.

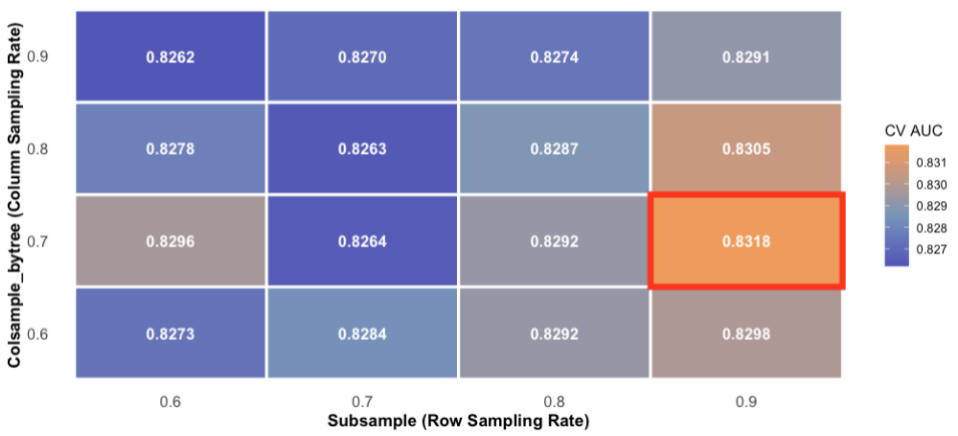

Figure 3: Subsampling Grid Search

Stage 3: Figure 3 shows that 16 combinations are tuned together from row subsampling (subsample) and column subsampling (colsample_bytree). Subsampling is a form of encoded regularization done by training each tree on random subsets of observations and features. Optimal values are subsample = 0.7 and colsample_bytree = 0.8.

Stage 4: The minimum child weight and minimum loss reduction for splits (gamma) are altered over 0, 0.1, 0.2, 0.5 × 1, 3, 5, 7. In order to prevent overfitting, these parameters are split to produce significant improvements and leaves to contain enough observations. The final values are min_child_weight = 5 and gamma = 0.1.

Hyperparameter tuning reaches a cross-validation AUC of 0.828 (SE = 0.002, 95% CI: [0.825, 0.831]), with consistent performance across all folds. Similar to the baseline model, this model's decision threshold is optimized to maximize F1 score on the test set. This yields an optimal threshold of 0.31, which reflects the class imbalance (25% high adoption).

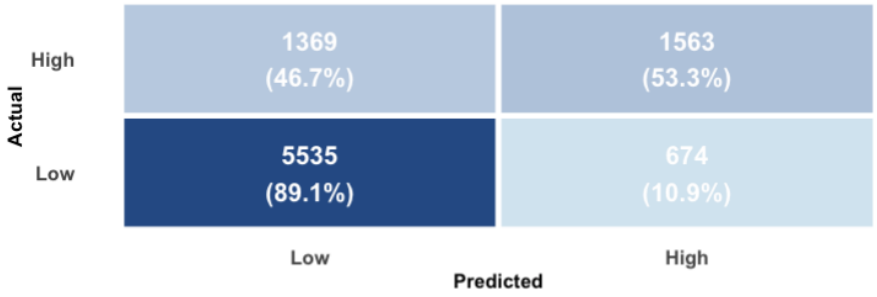

Figure 4: XGBoost Confusion Matrix

At the optimal threshold, XGBoost achieves a reliable prediction between high and low adoption models with a test set AUC of 0.835, and an F1 score of 0.605. Recall at 0.699 exceeds precision (0.533), suggesting that the model captures most high adoption models but at the cost of more frequent false positives. (Figure 4) This asymmetry reflects the threshold choice, when threshold was set at 0.31 to

maximize F1 given class imbalance, the model favors sensitivity over specificity. The practical implication is that browse-time data can screen for likely adoption successes but cannot reliably confirm them. Although, through the assessment of XGBoost's calibration, the model generated a Brier score of 0.159, well below the 0.1875 baseline expected from predicting the base rate (25%) for all observations, thus its predicted probabilities can reasonably approximate actual adoption rates. In other words, this model can be useful for narrowing options and its estimates can be trusted, but less so for definitive selection.

4.3 Evaluation and Comparison

Table 1: Classification Metric Comparison

Metric	Naive Bayes	XGBoost	Improvement
AUC	0.706	0.835	+18.3%
F1	0.480	0.605	+26.0%
Precision	0.375	0.533	+42.1%
Recall	0.666	0.699	+5.0%
Accuracy	0.647	0.777	+20.1%

XGBoost outperforms Naive Bayes across all metrics (Table 1), with AUC, F1, and precision increasing by 18%, 26%, and 42%, respectively. This performance gap confirms that feature interactions are essential for accurate adoption prediction. The F1 score for both models are on the lower end, since given that the 75th percentile adoption threshold crerates a 3:1 ratio, high precision and recall are difficult to achieve simultaneously. Though an F1 score of 0.61 from gradient boosting under this imbalance does imply that the model achieves a balanced trade-off, but accurate identification of high adoption models at the individual level remains a challenge. The numbers are also evidence that developers may rely substantially on unobserved data, such as documentation quality, community, or personal recommendations, thus information assymetry persists even when transparent model data is available.

4.4 Global Feature Importance

Mean absolute SHAP values quantify each feature's average contribution to prediction magnitude in the boosting model across all test observations. The top 15 features by importance (Figure 5) are:

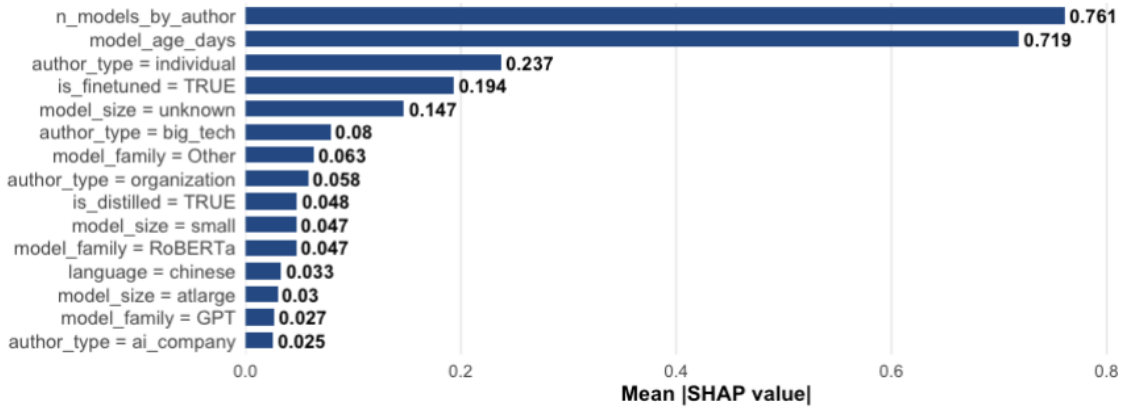

Figure 5: Top 15 Features by SHAP Importance

Author productivity (*n_models_by_author*) is one of the strongest predictors by far, with mean absolute SHAP value of 0.761, which is nearly three times larger than any categorical indicator, demonstrating that the accumulated track record of a model creator even trumps the author type in importance (individual, organization, etc). Model age ranks second (0.719), likely due to both longer exposure time and the buildup of downloads across longer observed periods, though a limitation exists as it is difficult to determine if model age indicates quality, visibility, or simply more time to accumulate downloads. Together, the two variables explain the majority of the model's predictive importance, incdicating that adoption is driven primarily by a model's cumulative exposure and the contributor's track record instead of intrinsic model descriptors.

Among categorical features, the individual developer indicator contributes most strongly (0.237), followed by fine-tuning status (0.194). The *big_tech* indicator, despite its strong association with high adoption in descriptive statistics, ranks only sixth in SHAP importance (0.080), suggesting that much of the big tech advantages are through correlated features, particularly with the high and continuous productivity of corporate research teams.

Closer examination of adoption rates by author category provides additional evidence of systematic advantages:

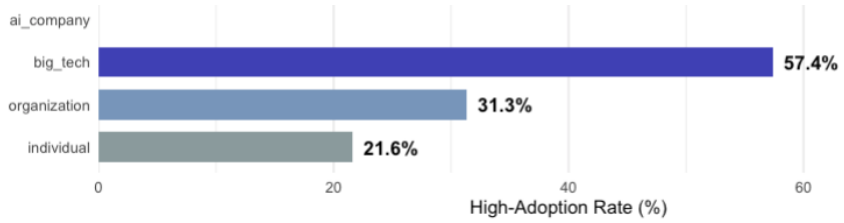

Figure 6: Adoption rates by author type

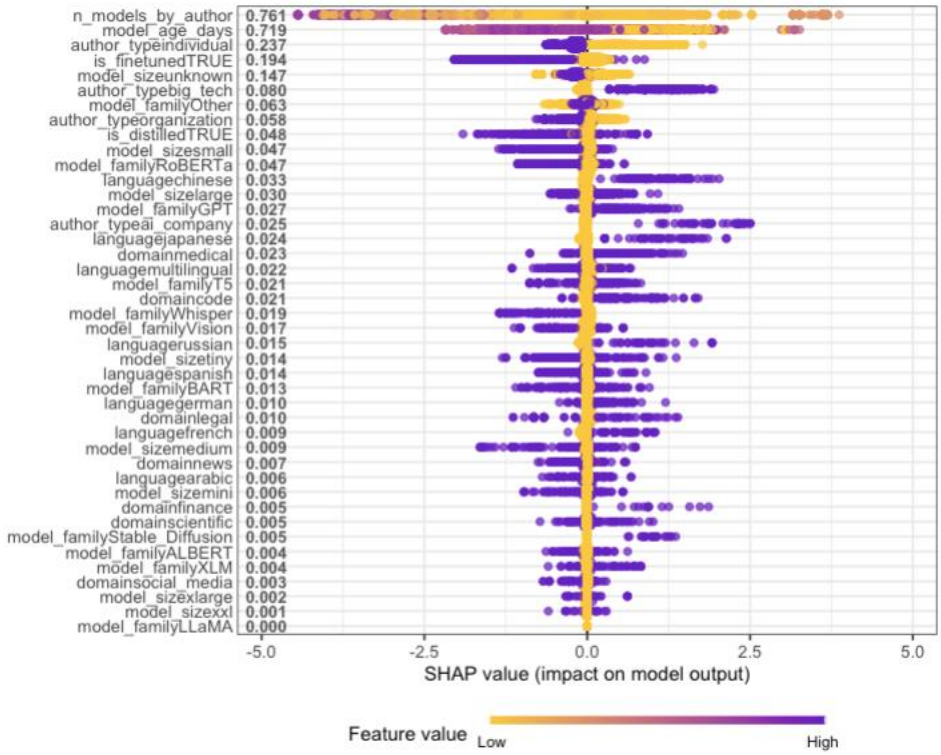

Figure 7: Adoption rates by author type

Figure 6 shows that big tech companies achieve the highest adoption rate at 57.4%, followed by organizations (31.3%), and individual developers (21,6%), despite individuals contributing 82.8% of all models in the sample. Figure 7 reinforces this pattern: *author_type_individual* ranks third in SHAP importance and is associated with SHAP values that vary across observations, while big tech (*author_type_big_tech*) is more consistently associated with positive SHAP values.

Conclusion

This study aims to predict adoption of AI models on Hugging Face using platform signals observable at browse-time. Analysis with Naïve Bayes and XGBoost yield two main findings:

First, adoption is indeed predictable from observable features, with indications that feature interactions and their non-linear relationship play an important role. Gradient boosting correctly ranks high adoption models above low adoption models 83.5% of the time, confirming that the prediction is not random. The result subtly implies that, on open-platforms such as Hugging Face, rather than directly evaluating the model quality, users may rely on easily observable indicators to assist in selection. Research also shows that, when technical quality assessment is costly, adopters rely on simpler proxies to reduce uncertainty. (Connelly et al., 2011) The 9 signals used in this study are enough to forecast outcomes with an acceptable degree of accuracy, however, precision of 0.533 indicates that individual-level prediction is still challenging, indicating that some drivers of adoption are not captured by the immediate data.

Second, the SHAP importance plot of the XGBoost model (global interpretation) reveals that technical browse-time variables play a secondary role to adoption prediction, as the model and contributors' cumulative and reputational signals dominate the feature importance. This phenomenon implies a self-reinforcing diffusion process in open platforms, where regardless of model characteristics, early visibility attracts more downloads by lowering uncertainty for prospective users (Perc, 2014) Even in the absence of evident preferential treatment by the platform, this generates a cumulated advantage because active contributors capture attention, then in turn, becomes "more reliable" and attracts more attention. Author type is an example of this, where individual users, the majority of Hugging Face model's origin, lowers predicted adoption, while big tech affiliation (a reputation signal itself) shows more positive adoption results, meaning that institutional backing may serve as credibility that reduces the initial perceived risk.

Several limitations call for acknowledgment. Technical characteristics are measured through proxies extracted from model titles rather than directly from the AI model, potentially resulting in minor measurement errors. This research design cannot establish any causality, since, for example, organizational signals may predict high adoption merely because they represent genuine model quality advantages instead of reputation. While defensed in the data section, the 75th percentile threshold cutoff to categorize "high adoption" is somewhat an arbitrary cutoff within a continuous distribution of download counts. Download counts also only measure initial adoption decisions, and does not reflect any behaviors from long-term usage, so a model with high downloads could be abandoned if it underperforms. Lastly, the Kaggle sample captures only a snapshot of a rapidly evolving platform, so a bigger sample size could benefit the results, and patterns might also shift as the ecosystem continues to develop.

Future research could examine how open platform designs, such as Hugging Face, shapes AI diffusion trajectories. Platforms implicitly bias certain signals over others when they decide which to showcase at browse time, therefore, while emphasizing creator history may reduce adoption risk for users, it can also lead to sole concentration on established players, limiting the exploration of new models. To better understand how these platforms should align diffusion driven by reputation or evaluative merit, other discovery variables and mechanisms should also be explored.

References

Abdaali, Y. (2023). *Hugging Face models dataset* [Data set]. Kaggle. https://www.kaggle.com/datasets/yasirabdaali/hugging-face-models-dataset

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., & Drame, M. (2020). *Transformers: State-of-the-Art Natural Language Processing.* Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Parker, G. G., & Van Alstyne, M. W. (2005). *Two-Sided Network Effects: A Theory of Information Product Design. Management Science*, 51(10), 1494–1504.

Lerner, Josh, and Jean Tirole. “Some Simple Economics of Open Source.” *The Journal of Industrial Economics*, vol. 50, no. 2, 27 Mar. 2003, pp. 197–234, https://doi.org/10.1111/1467-6451.00174.

Dabbish, Laura, et al. “Social Coding in GitHub.” *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work - CSCW ’12*, 2012, https://doi.org/10.1145/2145204.2145396.

Perc, Matjaž. “The Matthew Effect in Empirical Data.” *Journal of the Royal Society Interface*, vol. 11, no. 98, 6 Sept. 2014, p. 20140378, https://doi.org/10.1098/rsif.2014.0378.

Connelly, Brian L., et al. “Signaling Theory: A Review and Assessment.” *Journal of Management*, vol. 37, no. 1, 20 Dec. 2011, pp. 39–67.