Will Your Paper Get Accepted?
Predicting NeurIPS Submission Success

Lorenzo Emer Riccardo Porcedda

PhD AI for Society – Sant'Anna School of Advanced Studies, Pisa
Course: Statistical Learning and Large Data (SLLD) - prof. Francesca Chiaromonte

May 19, 2025

Abstract

Predicting the acceptance of academic papers is an increasingly relevant task in the context of highly competitive conferences such as NeurIPS. In this work, we develop a classification pipeline to predict the acceptance of submissions to the NeurIPS 2024 Conference using only features available at submission time, including structural characteristics, metadata, and full-text content represented via TF-IDF. We address the high dimensionality of text by applying SVD, and compare multiple classification models, including LASSO-regularized logistic regression, tree-based models, and support vector machines. Our most interpretable - yet one of the best - model achieves an F1-score of 0.898, stating that it is indeed possible to find patterns and biases that lead to acceptance or rejection. This model reveals that papers with clear visual structure (e.g., many figures and tables) and cutting-edge topics (e.g., graph generation, llava) are more likely to be accepted, while excessive lexical diversity may be penalized. We also discuss the limitations of using single-year data and the lack of author metadata for rejected papers, proposing future directions that include deeper network-based features and cross-year generalization.

1 Introduction

NeurIPS is one of the most prestigious conferences in machine learning and artificial intelligence, with acceptance rates typically below 26% [1]. For researchers, anticipating the likelihood of acceptance can inform both content strategy and submission timing, while for conference organizers, early predictions may assist in reviewer assignment or quality screening.

While peer review is inherently subjective and multifaceted, recent studies have attempted to model acceptance decisions using supervised learning techniques. These efforts are limited by small datasets, narrow feature sets, or a reliance on reviewer comments, which are not available at submission time. In this work, we address two research questions:

- **RQ1: Can we accurately predict the acceptance of NeurIPS submissions using features derived from the paper text and structure?**
- **RQ2: Which textual or structural features are most influential in determining the acceptance of a NeurIPS submission?**

To answer these questions, we build a dataset of 4,238 NeurIPS 2024 submissions scraped from OpenReview, and extract a rich set of features from the full-text, document structure, and metadata. We then apply dimensionality reduction, feature selection, and a wide range of classification models, evaluating their ability to generalize to unseen data. Our aim is not only to achieve strong predictive performance but also to understand which features most influence acceptance. Ultimately, this work contributes to the growing literature on peer-review prediction and reveals the important role of content (relatedness to ”in vogue” ML-themes such as Large Language Models and Graph Neural Networks) and structural clarity in determining success.

2 Literature Review

The literature on predicting the success of academic paper submissions to conferences remains relatively limited but is gradually expanding. [11] utilized the PeerRead dataset, applying various classifiers with engineered features and ten keywords extracted from abstracts. Their Random Forest model achieved an accuracy of 81%. Similarly, [6] focused on the ICLR 2017 dataset, extracting features such as the number of references, figures, tables, and the frequency of machine learning-related terms. They experimented with multiple classification models, including Logistic Regression, Decision Trees, Random Forests, KNN, and SVM. Decision Tree achieved a precision of 0.93, recall of 0.82, and an F1-score of 0.87 on their data, indicating a strong predictive capability of tree based-models. Other authors employed different approaches.

[8] investigated the use of sentiment analysis on reviewer comments to predict acceptance decisions. Utilizing a dataset comprising 2,313 review texts from two computer science conferences, they applied seven machine learning algorithms for both regression and classification tasks. SVM and MLP Classifier achieved the best performance in classification tasks, while Nearest Neighbors excelled in regression tasks. However, the authors recognized important limitations in employing sentiment analysis for these tasks, as most reviews were constructively written, leading to predominantly positive classifications and reducing the tool's effectiveness in detecting negative sentiments. [15] proposed a novel method combining individual and network features to predict paper acceptance rates. They calculated affiliation scores, constructed institutional collaboration networks, and analyzed the importance of institutions using network centrality measures. Their Random Forest algorithm surpasses other state-of-the-art methods. Overall, the state of the art reveals a lack of classification methods that consider all words in the paper (e.g., a full-text or high-dimensional model) to predict acceptance.

3 Data Description

The dataset contains 4,238 full paper submissions to NeurIPS 2024, scraped from the OpenReview platform. Of these, 4,037 were accepted and only 201 were rejected, prospecting a highly imbalanced binary classification problem with a strong skew toward the positive class (accepted articles represent approximately 95.3% of the total). For each submission, we extracted features from three sources: the manuscript text, the document structure and the associated metadata.

Textual features include: (i) total word count, (ii) count of unique words (i.e. the vocabulary adopted) and (iii) average sentence length. Structural features comprise: (i) number of figures, (ii) number of tables, (iii) number of equations , and (iv) number of references. Metadata features include: (i) length of the title (in words), (ii) length of the abstract (in words) and (iii) a binary indicator which indicates whether supplementary material was submitted. In Table 1 we show a synthetic table of all these extracted features.

This feature engineering was made after a thorough preprocessing of the texts which is described in Section 4.

4 Methodology

The methodology we followed comprised the following steps (also depicted in Figure 1): (i) scraping, (ii) preprocessing and feature engineering, (iii) construction of the TF-IDF matrix, (iv) Singular Value Decomposition, (v) training of classification models, (vi) evaluation and (vii) interpretation of feature importance.

Category	**Feature Description**
Textual	Total word count Count of unique words Average sentence length
Structural	Number of figures Number of tables Number of equations Number of references
Metadata	Title length Abstract length Supplementary material (bool)

Table 1: Overview of extracted features from NeurIPS 2024 Papers.

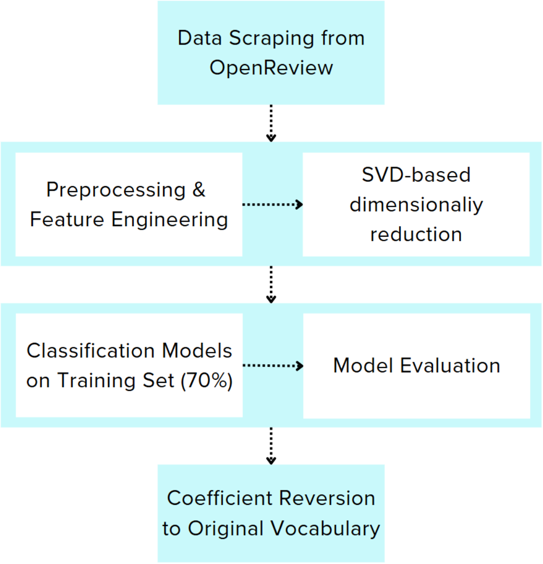
Data Scraping from OpenReview
Preprocessing & Feature Engineering
SVD-based dimensionaliy reduction
Classification Models on Training Set (70%)
Model Evaluation
Coefficient Reversion to Original Vocabulary

Figure 1: Graphical illustration of the employed methodology

i) Scraping. To collect the dataset, we scraped all publicly available submissions to NeurIPS 2024 from the OpenReview platform. The scraping process was implemented in Python, combining several libraries for automated browsing and structured data extraction. Specifically, we used:

- `Selenium` for browser automation and interaction with dynamic content,
- `BeautifulSoup` for parsing and extracting relevant HTML elements (e.g., paper titles, abstracts, acceptance decisions, PDF links),
- `Pandas` and `NumPy` for storing, manipulating, and exporting the extracted data.

ii) Preprocessing and Feature Engineering. The PDFs downloaded using the scraped web links, were then parsed using the `PyPDF2` library. We proceeded then with segmentation of the text into main content, appendix, and references. Specifically, we located and removed the "NeurIPS Paper Checklist" section, as well as performed regular-expression-based searches for "Appendix" headings or enumerated references. Also, in order to create a dataset with no explicit indication of paper acceptance/rejection, we removed the author section, the corresponding authors, the acknowledgments and the sentences *Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.* and *38th Conference on Neural Information Processing Systems (NeurIPS 2024).* We also searched for regular expressions (RegEx) indicative of tables, figures, or mathematical expressions. For math detection, our code combined sets of extended Greek letters, symbolic characters, and common function keywords (e.g., "`sin`", "`log`"). Finally, the features obtained are the one in Table 1.

iii) Construction of TF-IDF Matrix. A TF-IDF matrix (Term Frequency-Inverse Document Frequency matrix) [12] is a numerical representation of a collection of documents, indicating the importance of each term in a document relative to the entire corpus. The TF-IDF score for a term t in a document d is the product of two components:

$$TF(t,d) = \frac{\text{Appearances of } t \text{ in document } d}{\text{\#terms in } d}$$

$$IDF(t) = \log\left(\frac{N}{1 + \text{\#documents containing } t}\right)$$

where N is the total number of documents in the corpus. The result is

$$\text{TF-IDF}(t,d) = TF(t,d) \times IDF(t)$$

which is a $n \times d$ real-valued matrix This highlights important terms in each document, but reduces the influence of common terms that appear in many documents.

iv) Singular Value Decomposition. To reduce the dimensionality of the TF-IDF matrix while preserving its most informative structure, we apply Singular Value Decomposition (SVD).

SVD factorizes the matrix as:

$$\mathbf{T} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^\top,$$

where:

- $\mathbf{U} \in \mathbb{R}^{n\times q}$ contains the top q left singular vectors (document embeddings),
- $\mathbf{\Sigma} \in \mathbb{R}^{q\times q}$ is a diagonal matrix of the top q singular values, and
- $\mathbf{V} \in \mathbb{R}^{d\times q}$ contains the corresponding right singular vectors (term embeddings).

Each document (or paper) is thus represented by its projection onto the top q latent semantic dimensions:

$$\mathbf{u}_i = \big[\mathbf{U}\big]_{i,\cdot} \in \mathbb{R}^q.$$

For the classification task, we concatenate this reduced representation $\mathbf{u}_i$ with the numerical features in Table 1.

The dimension of $\mathbf{U}$ can be further reduced if we consider the low-rank approximation of the TF-IDF matrix

$$\mathbf{T} \simeq \mathbf{U}_k\,\mathbf{\Sigma}_k\,\mathbf{V}_k^\top,$$

using only the first $k < q$ singular components.

Once a (linear) regression model is estimated using the reduced latent dimensions $\mathbf{U}_k$ [2], we wish to interpret the results in terms of the original TF-IDF features. Let be $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^k$ the coefficients of the regression on $\mathbf{U}_k$, then the corresponding coefficients on the original feature space are given by:

$$\hat{\boldsymbol{\beta}} = \mathbf{V}_k\mathbf{\Sigma}_k^{-1}\hat{\boldsymbol{\alpha}},$$

which recovers an estimate of the impact of each original term (column of $\mathbf{T}$) on the outcome variable. Similarly, standard errors can be transformed using the same linear mapping: the covariance matrix of $\hat{\boldsymbol{\beta}}$ is given by

$$\text{Var}(\hat{\boldsymbol{\beta}}) = \mathbf{V}_k\mathbf{\Sigma}_k^{-1}\text{Var}(\hat{\boldsymbol{\alpha}})\mathbf{\Sigma}_k^{-1}\mathbf{V}_k^\top.$$

This transformation allows inference to be conducted in the original term space, even though estimation was carried out in the lower-dimensional latent semantic space, providing both computational efficiency and interpretability.

6) Train/Test Split. We randomly partition the dataset into a training set $\mathcal{D}_{\text{train}}$ (70%) and a test set $\mathcal{D}_{\text{test}}$ (the remaining 30%), ensuring the same fraction of positive and negative examples. Let N_{train} denote the number of papers in the training set.

7) Classification Models To predict paper acceptance, we employed a variety of classification algorithms spanning both linear and non-linear paradigms. Our baseline was **logistic regression with LASSO regularization**, which is both interpretable and capable of performing embedded feature selection [13]. It estimates the probability that a paper i is accepted as:

$$P(y_i = 1 \mid \mathbf{x}_i) = \frac{1}{1 + \exp(-\mathbf{x}_i^\top\boldsymbol{\beta})},$$

where $\mathbf{x}_i \in \mathbb{R}^p$ is the feature vector, and $\boldsymbol{\beta}$ are the model coefficients. LASSO adds an L1 penalty to enforce sparsity:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{\mathcal{L}(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1\right\},$$

where $\mathcal{L}(\boldsymbol{\beta})$ is the logistic loss and λ controls the strength of regularization.

We further explored the following models [5]:

- **Decision Trees [7]:** recursively split the feature space to minimize node impurity, defined as $\sum_k p_k(1-p_k)$ over class proportions p_k.
- **Random Forests [3]:** ensembles of decision trees trained on bootstrapped samples and random subsets of features, with majority voting to reduce variance.
- **Support Vector Machines (SVM) [11]:** optimize the margin between classes by solving

  $$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1,$$

  extended to soft margins and kernel methods for non-linear separation.
- k**-Nearest Neighbors (k-NN) [4]:** non-parametric model classifying a point based on the majority label among its k nearest neighbors. Best results were achieved with $k = 5$.

To address class imbalance and high-dimensionality, we tested several **variants of logistic regression**:

- **Class-weighted logistic regression:** applies weights w_i inversely proportional to class frequency in the loss function [16].
- **SIS + LASSO:** combines Sure Independence Screening [10] to pre-select features with LASSO for final selection.
- **SIS only:** trains logistic regression directly on SIS-selected features.
- **Stability-based downsampling:** selects only features stable across multiple LASSO runs (present in >20% of 20 subsamples) [14].

8) Model Evaluation To assess the predictive performance of our classification models, we employed several standard metrics derived from the **confusion matrix**, which tabulates true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These metrics allow us to quantify different aspects of classification performance, particularly in the presence of class imbalance [5].

Precision measures the proportion of predicted positives that are true positives, indicating how reliable positive predictions are:

$$\text{Precision} = \frac{TP}{TP + FP}.$$

Recall (also called sensitivity or true positive rate) captures the proportion of actual positives that are correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN}.$$

F1-score is the harmonic mean of precision and recall, balancing the two when there is a trade-off:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

In addition to threshold-based metrics, we considered ranking-based metrics that are more appropriate in settings with high class imbalance. The **Area Under the ROC Curve (AUC)** [8] evaluates how well a model ranks positive instances ahead of negative ones. It is equivalent to the probability that a randomly chosen positive example ranks higher than a randomly chosen negative one.

We also used **Average Precision (AP)**, which summarizes the precision-recall curve by averaging precision across different recall levels. Unlike accuracy, AP is sensitive to the quality of ranking, making it particularly useful when the minority class is of greater interest (e.g., rejected papers) [9]:

$$\text{AP} = \sum_n (R_n - R_{n-1})P_n,$$

where P_n and R_n are the precision and recall at the n-th threshold.

Together, these metrics provide a comprehensive indication of each model's discriminative ability and robustness to imbalance.

5 Results

Classification Performance We evaluated all models on a held-out test set, using standard classification metrics: Precision, Recall, F1-score, Area Under the ROC Curve (AUC), and Average Precision (AP), as described by Table 5. Among all methods, the **class-weighted logistic regression** emerged as the best overall performer, offering a compelling balance between accuracy and interpretability. It achieved an F1-score of 0.898, AUC of 0.813, and AP of 0.987, correctly predicting 1003 out of 1210 accepted papers as well as 41 out of 61 rejected papers, therefore demonstrating robustness to class imbalance. A confusion matrix for some selected models is provided in Table 5.

The baseline **logistic regression with LASSO** also performed well, achieving an F1-score of 0.840, AUC of 0.677, and AP of 0.975. A retrained version using only statistically significant variables achieved slightly lower performance (F1 = 0.827, AUC = 0.672), but benefited from a more concise and interpretable set of predictors.

Among the more exploratory strategies, **SIS + LASSO** failed to generalize and collapsed to a single-class prediction. On the other hand, the simpler **SIS-only** variant achieved high recall and F1-score (0.924), but suffered from weak discriminative ability, as indicated by its low AUC of 0.630.

Model	**F1-Score**	**AUC**	**AP**	**Precision**	**Recall**
Logistic Regression (LASSO)	0.840	0.676	0.975	0.966	0.744
Logistic Regression (Significant Vars)	0.827	0.672	0.974	0.968	0.721
Weighted Logistic Regression	**0.898**	**0.813**	**0.987**	**0.980**	0.829
SIS + LASSO	N/A	N/A	N/A	N/A	N/A
SIS Only	0.924	0.630	0.969	0.957	0.893
Downsampling (Stable Vars)	N/A	N/A	N/A	N/A	N/A
Decision Tree (Weighted)	0.972	0.595	0.959	0.956	**0.988**
Random Forest (Weighted)	**0.975**	0.534	0.954	0.952	**1.000**
SVM (Weighted)	0.974	0.804	**0.987**	0.953	0.997
k-NN (K=5)	0.973	0.607	0.963	0.952	0.996

Table 2: Performance of Classification Models on NeurIPS 2024 Submissions

Tree-based classifiers delivered mixed results. The **decision tree** model reached high recall (0.988) and F1-score (0.972), but failed in specificity, correctly identifying only 3 out of 76 rejected papers (AUC = 0.595). The **random forest** displayed a similar pattern: near-perfect recall but minimal class separation, yielding a low AUC of 0.534. These models tended to overfit on the majority class, reducing their practical utility despite high recall.

Finally, the **Support Vector Machine (SVM)** achieved strong performance across the board, with an F1-score of 0.974, recall of 0.997, and AUC of 0.804. However, like other high-capacity models, it suffers from poor interpretability. The **k-Nearest Neighbors (k-NN)** classifier also performed well (F1 = 0.973), though its lower AUC (0.607) suggests limited ability to rank examples correctly under severe class imbalance.

Overall, the best results were obtained by combining regularized linear models with appropriate class weighting.

Feature-level Analysis Beyond predictive performance, one of the strengths of the LASSO-regularized logistic regression lies in its interpretability. The model produced a sparse set of coefficients that highlight the features most associated with paper acceptance. Among the structural variables, the number of figures ($\hat{\beta} = 0.1525$) and total word count ($\hat{\beta} = 3.81 \times 10^{-4}$) were strong positive predictors, suggesting that longer submissions with substantial visual content are more likely to be accepted. In contrast, unique word count had a negative effect ($\hat{\beta} = -0.00317$), which may reflect a reviewer preference for focused, less lexically diverse writing (Table 4).

When retraining logistic regression on only the statistically significant features, these trends were further reinforced. Additional variables such as the number of tables ($\hat{\beta} = 0.1666$), number of figures ($\hat{\beta} = 0.2563$), and average sentence length ($\hat{\beta} = 0.043$) all contributed positively. Meanwhile, the negative coefficient on unique word count persisted, emphasizing the importance of structural clarity and consistent terminology in successful submissions.

On the textual side, LASSO also selected several high-weight TF-IDF components, such as `V2`, `V590`, `V715`, and `V807`, which encode latent textual patterns. After reversing the SVD projection, we identified the most influential underlying terms, based on both coefficient magnitude and statistical significance, as cutting-edge machine learning topics, including `llava`, `graph_generation`, and `actionvalue_function` (Table 5). The relevance of these keywords suggests that content affiliation to emerging or cutting-edge research areas is a strong driver of acceptance.

6 Discussion

Among all evaluated models, **Weighted Logistic Regression** offered the best trade-off between predictive performance and interpretability, achieving an F1-score of 0.90 and an AUC of 0.81. This model not only performed well under severe class imbalance but also provided meaningful insights into the factors associated with paper acceptance at NeurIPS.

The most influential **positive signals** included structural features such as the number of tables and figures, suggesting that reviewers may favor submissions that characterized by visual clarity and evidence-based presentation. Additionally, a moderate positive effect was observed for average sentence length, possibly reflecting more elaborate or well-articulated writing styles.

Conversely, **unique word count** emerged as a negative predictor, which may indicate that excessive lexical diversity or lack of focus in terminology is penalized during peer review. This finding aligns with the intuition that clarity and conciseness are valued over novelty in vocabulary.

At the word-level, several high-weighted TF-IDF components corresponded to cutting-edge machine learning topics, including `llava`, `graph_generation`, and `actionvalue_function`. The presence of such domain-specific keywords among top predictors highlights the influence of topical relevance and trends

Model	Actual = 0, Pred = 0	Actual = 0, Pred = 1	Actual = 1, Pred = 0	Actual = 1, Pred = 1
Logit + LASSO	32	29	337	873
Weighted Logit	41	20	207	1003
SIS (No LASSO)	12	49	129	1081
SVM	1	60	4	1206

Table 3: Confusion matrices for selected models on the test set. Each row shows the number of predictions by class (0 = Rejected, 1 = Accepted).

LASSO Logistic Regression Model

Variable	Coefficient
word_count	0.000381
unique_word_count	-0.003171
num_figures	0.1525
V2	51.40
V590	15.70
V715	-19.54
V807	18.55
V1876	-22.51
V2184	19.99

Logistic Regression on Significant Variables

Variable	Coefficient
word_count	0.000919
unique_word_count	-0.006665
num_tables	0.1666
num_figures	0.2563
avg_sentence_length	0.04305
V2	184.0
V3	-51.82
V18	-36.11
V46	-45.72
V132	-36.26
V145	-35.85
+ 22 additional TF-IDF dimensions	

Table 4: Selected coefficients from logistic regression models. The LASSO model retained 6 TF-IDF features; the retrained model retained 39 TF-IDF features plus structural/textual variables.

Top 10 by Coefficient ($\hat{\beta}$)	Bottom 10 by Coefficient ($\hat{\beta}$)	Top 10 by p-value	Bottom 10 by p-value
ace	simplex	handle_case	graph_generation
actionvalue_function	free	type_error	llava
dynamic_scene	cat	open_vocabulary	separation
graph_generation	scan	initialize_parameter	blurring
blurring	reaction	parallel_computing	delay
llava	probability_mass	latent_dynamic	free
overlap	separation	independent_run	dynamic_scene
pip	sequential	uncertain	sequential
emp	ope	cooccurrence	amplitude
request	amplitude	phase	ope

Table 5: TF-IDF Terms: Top and Bottom by Coefficient and p-value

in determining acceptance. These results that technical content and presentation quality both have an important role in shaping reviewer decisions.

We further highlight a peculiar finding: when including acknowledgment text (without the "Acknowledgment" heading, which would be a perfect predictor of acceptance), we would think that a rich set of grants and fundings would constitute a set of predictors of papers acceptance. This was not the case: in fact, just one grant (*National Natural Science Foundation of China (NSFC)*) was depicted by the model as a positive predictor.

7 Limitations and Future Work

While our study demonstrates promising results in predicting NeurIPS paper acceptance using submission-time features, several limitations should be acknowledged. First, our analysis is restricted to a single year (NeurIPS 2024), which may limit the generalizability of the results across time or different venues. Future work should aim to incorporate data from other editions of NeurIPS and comparable machine learning conferences (e.g., ICML, ICLR, AAAI) to improve robustness and assess temporal consistency in acceptance criteria.

Second, author and affiliation information was only available for **accepted papers**, as NeurIPS conceals these details for rejected submissions. This constraint currently prevents us from conducting author-level or institution-level analyses across the full dataset. To enable network-based extensions (e.g., co-authorship or institutional influence), we would need either to link papers to authors using external data sources (e.g., a rejected paper could still be found as a pre-print in arXiv) or to develop heuristics for tracing back anonymized submissions to known publications (basically, a machine learning model could try to predict the authors of a paper based on content and references). Nonetheless, we could still analyze only accepted papers to check the dynamics of grants and fundings, in order to unveil influences of financing external entities in the scientific editorial world. This would expand the findings discussed in last paragraph of Section 6.

Third, our current modeling framework relies on classical machine learning methods. While these offer good interpretability and efficient training, they may not be able to fully capture complex interactions in the data. Exploring more advanced models such as deep learning architectures (e.g., transformers applied to full texts) or gradient boosting methods (e.g., XGBoost, LightGBM) could improve predictive performance, and therefore it represent a valuable expansion of this work.

8 Conclusions

This work demonstrates that it is possible to predict NeurIPS acceptance with decent accuracy using only submission-time information. Through a combination of TF-IDF-based text representation, SVD dimensionality reduction, and LASSO-regularized logistic regression, we achieved competitive performance while maintaining model interpretability. The best model (class-weighted logistic regression) balanced precision and recall effectively, with an F1-score of 0.898 and an AUC of 0.813.

Our analysis shows that papers featuring more tables and figures, longer sentences, and concise vocabularies are more likely to be accepted. Moreover, full-text terms related to cutting-edge topics such as `llava`, `graph_generation`, and `actionvalue_function` emerged as strong positive signals. These findings underscore how both content and presentation influence reviewer decisions.

Despite these promising results, the study is constrained by its focus on a single year and the lack of complete metadata for rejected papers. Future work should extend this approach to multiple years and conferences, incorporate author-level network features, and explore modern NLP architectures for richer text representations.

References

[1] 38th annual conference of neural information processing systems fact sheet. 2024.

[2] John Mandel and. Use of the singular value decomposition in regression analysis. *The American Statistician*, 36(1):15–24, 1982.

[3] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[4] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009.

[6] D. J. Joshi, A. Kulkarni, R. Pande, I. Kulkarni, S. Patil, and N. Saini. Conference paper acceptance prediction: Using machine learning. In A. Dutta, L. C. Jain, and S. Aggarwal, editors, *Machine Learning and Information Processing: Proceedings of ICMLIP 2020*, pages 143–152. Springer Singapore, 2021.

[7] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[8] A. C. Ribeiro, A. Sizo, H. Lopes Cardoso, and L. P. Reis. Acceptance decision prediction in peer-review through sentiment analysis. In G. Marreiros, F. S. Melo, N. Lau, H. Lopes Cardoso, and

L. P. Reis, editors, *Progress in Artificial Intelligence. EPIA 2021*, volume 12981 of *Lecture Notes in Computer Science*, pages 757–769. Springer, Cham, 2021.

[9] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

[10] Daniel F. Saldana and Yang Feng. SIS: An R Package for Sure Independence Screening in Ultrahigh-Dimensional Statistical Models. *Journal of Statistical Software*, 83(2):1–25, 2018.

[11] M. Skorikov and S. Momen. Machine learning approach to predicting the acceptance of academic papers. In *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pages 113–117. IEEE, 2020.

[12] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[13] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[14] Haoda Wang, Rong Zhu, and Ping Ma. Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, 113(522):829–844, 2018.

[15] W. Wang, J. Zhang, F. Zhou, P. Chen, and B. Wang. Paper acceptance prediction at the institutional level based on the combination of individual and network features. *Scientometrics*, 126:1581–1597, 2021.

[16] J. R. Wilson, K. A. Lorenz, and L. P. Selby. Weighted logistic regression model. In *Modeling Binary Correlated Responses: Using SAS, SPSS, R and STATA*, pages 85–106. Springer International Publishing, Cham, 2024.