Proceedings of CONF-CDS 2025 Symposium: Data Visualization Methods for Evaluation
DOI: 10.54254/2755-2721/2025.PO24816

Predicting User Preferences in LLMs: A Real - World Data - Driven Approach Based on RLHF

Zihao Peng[1†], Qixiao Gao[2*†], Ruiyan Li[3], Yannan Li [4], Qixiao Wang[5]

[1]*School of Mathematical and Statistical Sciences, Arizona State University, Tempe, USA*
[2]*School of Electrical and Computer Engineering, Xiamen University Malaysia, Salak Tinggi, Malaysia*
[3]*School of Business Administration, Southwestern University of Finance and Economics, Chengdu, China*
[4]*School of Environmental Science and Engineering, Shanghai Jiao Tong University, Shanghai, China,*
[5]*Haidian Foreign Language School, Beijing, China*
**Corresponding Author. Eamil: 2324235225@qq.com*
[†]These authors contributed equally to this work and should be considered as co-first author.

Abstract. Aligning large language model (LLM) outputs with human preferences remains challenging, as traditional metrics like perplexity fail to capture multi-dimensional user satisfaction. This paper proposes a data-driven reinforcement learning from human feedback (RLHF) approach that integrates flipping-based data augmentation and ensemble learning to optimize user preference prediction. Leveraging a multilingual dataset from the Chatbot Arena platform (containing user prompts, paired LLM responses, and human evaluations), we design a flipping augmentation strategy that generates mirrored samples by swapping response positions and inverting labels, thereby balancing dataset distribution and mitigating position bias. Furthermore, by extracting statistical text features (e.g., length, punctuation frequency, language ratios) and combining gradient-boosted models (CatBoost, LightGBM, XGBoost), we build a stacked ensemble model with logistic regression. Threshold tuning (optimal cutoff: 0.503) elevates accuracy to 60.2%. Experiments demonstrate a 3.2% generalization improvement on unflipped test sets, validating the effectiveness of position bias mitigation. This study provides a systematic framework for RLHF deployment in multilingual environments and highlights future challenges related to complex biases and computational efficiency.

Keywords: Large Language Models (LLMs), RLHF, User Preference Prediction, Data Augmentation.

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.PO24816

1. Introduction

Large Language Models (LLMs) now play a central role in reshaping human-computer interaction, powering innovations from automated customer support to intelligent information systems [1]. While these models demonstrate remarkable capabilities in scale and performance, a critical challenge persists: their outputs often fail to align with what humans genuinely prefer. Traditional metrics like perplexity, though effective for measuring text coherence, struggle to evaluate complex user satisfaction—especially in open-ended dialogues where responses must balance accuracy, contextual relevance, and social appropriateness [2]. For example, a response may be grammatically perfect yet socially tone-deaf—a nuance purely algorithmic metrics miss [3].

To tackle this limitation, researchers have turned to Reinforcement Learning from Human Feedback (RLHF), a paradigm where human evaluators rank model outputs to train reward systems [4]. By embedding real user preferences into training, RLHF steers LLMs toward responses that are not just factually correct but also contextually aware and socially aligned [5,6]. However, the quality of human feedback data directly determines RLHF's success. Even subtle biases—like users favoring the first response they see—can skew model behavior, leading to outputs that mirror these distortions [7,8].

In this study, we develop a preference prediction model using real-world data from Chatbot Arena, a platform where users compare pairs of LLM-generated responses to select a "winner." Our goal is to predict user preferences more accurately, thereby refining LLM alignment. Unlike previous approaches, we explicitly address three key issues that can bias user evaluations: position bias, where users tend to favor the first response regardless of its quality; verbosity bias, where longer responses may be preferred even when they are less relevant; and model identity bias, where evaluators might unconsciously favor outputs from well-known models.

To counter position bias, we introduce a symmetrical flipping strategy: by swapping response pairs and reversing labels (e.g., "winner" becomes "loser"), we double the dataset size (48k → 96k) and force models to focus on content rather than display order. For instance, if a sample originally ranks Response A above Response B, the flipped version ranks B above A. This approach not only balances positional dependencies but also reveals inherent response quality—a medical chatbot study found that 15% of preferences were driven purely by order (Rahman et al., 2024).

By combining this strategy with gradient-boosted models [9] and threshold optimization, our framework improves prediction accuracy while maintaining multilingual robustness. Ultimately, this work demonstrates how systematic data augmentation and model collaboration can address biases in real-world feedback loops. These advances pave the way for LLMs that better resonate with human values [10].

2. Literature review

Reinforcement Learning from Human Feedback (RLHF) uses human judgments to align LLM outputs with user preferences [4]. Early successes—such as instruction-following models [1] and helpful, harmless chatbots [10]—demonstrated this approach's promise. However, RLHF's effectiveness hinges on high-quality feedback data, and biases in human evaluations can skew the reward model.

Biases such as position bias (favoring the first response; [7]), verbosity bias (preferring lengthy answers), and model identity bias (crediting outputs from well-known models; [5]) can distort human feedback. For example, position bias swayed 15% of decisions in a medical chatbot study [9], and such effects may be amplified in multilingual contexts [11].

DOI: 10.54254/2755-2721/2025.PO24816

To mitigate these biases, researchers have tried randomizing answer order or modeling click behavior [7,8], though this becomes impractical at scale [11]. Verbosity bias has been tackled by retrieval-augmented generation to encourage concise answers, though this can constrain model flexibility, and by augmenting training data via paraphrasing at the risk of altering meaning [12]. In contrast, we introduce a flipping-based augmentation to neutralize position bias without altering content; each comparison is duplicated with responses swapped and labels inverted, doubling the data and forcing the model to focus on content rather than order.

Another gap is the limited use of ensemble learning and threshold calibration. Gradient boosting models like LightGBM and CatBoost excel with structured features [13]. For example, a LightGBM model using only text-length features achieved roughly 58% accuracy [14]. Combining classifiers via stacking can further improve robustness [15]. However, most studies still rely on a single classifier with a fixed 0.5 threshold, overlooking the benefits of model ensembles and threshold tuning [16].

In summary, current RLHF-based preference modeling faces three major challenges that our work addresses. First, position bias remains a persistent problem, as existing countermeasures such as randomizing response order tend to falter at scale, especially in multilingual environments. To overcome this, we propose a flipping-based augmentation strategy that swaps response positions and inverts labels, thereby balancing the presentation order and mitigating positional effects. Second, the integration of heterogeneous models has been underexplored; prior studies often rely on a single classifier, limiting robustness. To address this, we employ a stacked ensemble approach that combines CatBoost, LightGBM, and XGBoost via logistic regression, leveraging the strengths of multiple models to enhance prediction stability. Third, most existing methods adopt a fixed threshold of 0.5 for classification decisions, overlooking probabilistic nuances in human preferences. In response, we implement threshold tuning, identifying an optimal cutoff value of 0.503 to refine the decision boundary and boost predictive accuracy. By systematically tackling these gaps, our framework seeks to improve both the accuracy and fairness of user preference prediction in real-world LLM alignment tasks.

3. Data

The dataset used in this study comes from the 2025 WSDM Cup “Multilingual Chatbot Arena,” which was released for competition. It was collected and published by the ChatBot Arena project team, containing a large number of human evaluations comparing two different large language model (LLM) responses. Specifically, a user provides a prompt, two anonymous models generate separate answers (response_a and response_b), and a human rater selects the preferable answer based on factors such as accuracy, completeness, or readability. Compared with similar datasets, this one covers over a hundred languages and several dozen different LLM versions, providing a multilingual, multi-model, and highly diverse corpus.

The official dataset includes three files: train.parquet, test.parquet, and sample_submission.csv, with the latter two serving as references and examples. For our model training, we selected train.parquet as the core dataset (Table 1).

DOI: 10.54254/2755-2721/2025.PO24816

Table1: Fields and descriptions of dataset

Feature	Description
id	Unique identifier for each conversation
prompt	Text input from the user
response_a	The answer generated by model A
response_b	The answer generated by model B
winner	The preferred response as chosen by a human rater (“model_a” or “model_b”)
model_a	The model identifier for response A (visible only in the training set)
model_b	The model identifier for response B (visible only in the training set)
language	Language label (e.g., “English”, “Russian”), which may contain some noise or errors

Scale and Structure of the Dataset:

- Number of Samples The initial training set (train.parquet) has about 48,439 rows (8 columns). However, following a “flipping-based data augmentation” step, the number of training samples doubles to approximately 96,878.

- Multilingual Distribution In the training set, the language field covers more than 100 labeled languages, with English being the most common (about 50,422 entries), followed by Russian, Chinese, Vietnamese, and so on, each ranging from a few thousand to over ten thousand samples. The distribution follows a long-tail pattern, with some minority languages appearing only a few times.

Data Preprocessing and Feature Processing:

- Text Fields The prompt, response_a, and response_b fields may contain multiple languages, HTML or Markdown formatting, and other complexities. In subsequent feature engineering, we perform statistics on these fields (e.g., string length, punctuation counts, percentage of Chinese/English characters), but we do not heavily sanitize or remove content.

- Feature Engineering In the “Feature Creation” step, we extracted numerous statistical indicators for each text field (prompt, response_a, response_b), such as length, punctuation frequency, parenthesis matching, and digit/letter/Chinese proportions. These are then used in the downstream modeling phase.

To reduce potential bias toward the first-displayed response, we employ a symmetric flipping approach. Hence, each record (comprising a prompt, response_a, response_b, and a winner label) spawns a mirror record with the A/B responses swapped and the winner label inverted. Consequently, the training set size doubles from around 48k to 96k. This symmetrical approach helps the model learn the intrinsic quality of the answers instead of relying on position (e.g., always “A” first).

In conclusion, the training dataset (~48k samples, doubled to ~96k after flipping) spans over a hundred languages and numerous model versions. It also exhibits a perfectly balanced distribution of the 'winner’ label (50% 'model_a’ vs. 50% 'model_b’; see Figure 1). For a binary classification task, having nearly equal positive and negative cases—and eliminating strong dependence on position or specific model IDs—greatly reduces class imbalance and position bias. As a result, our model can focus more directly on learning and evaluating the actual quality of the answers. This provides a solid foundation for the subsequent preference-prediction model. We have also incorporated flipping-based augmentation to mitigate any positional bias, along with basic text-based feature extraction.

DOI: 10.54254/2755-2721/2025.PO24816

4. Method

In this study, we aim to predict which one of two Large Language Model (LLM) responses a user is more likely to prefer, based on real-world data collected from the Chatbot Arena platform. Our approach involves multiple stages, including data augmentation, feature engineering, training single models (CatBoost, LightGBM, XGBoost), building an ensemble model (Logistic Regression), and fine-tuning the classification threshold to maximize accuracy. Below, we describe each of these steps in detail (The overall workflow is shown in Fig.1.).

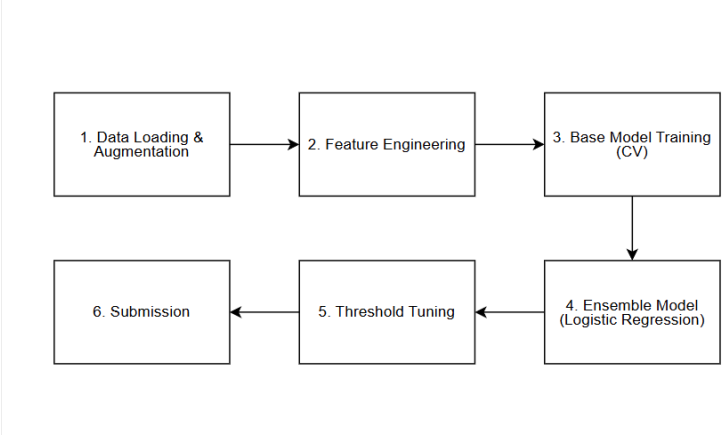

Figure 1: Overall workflow for predicting preferred LLM responses

4.1. Flipping-based augmentation

Each original record in "train.parquet" contains:prompt (the user's request),response_a and response_b (two AI-generated answers), winner (the answer chosen by a human reviewer, either "model_a" or "model_b").

To address position bias, we duplicate each sample but swap response_a with response_b and invert the winner label. After concatenation, we obtain a dataset of ~96,878 samples with nearly balanced labels (50% "model_a" vs. 50% "model_b"). This practice is based on the concept that user preferences are often influenced by the order in which answers are displayed, and flipping can symmetrically neutralize that advantage [7].

4.2. Feature engineering

We create a variety of numeric features from the text fields, including:

- Length & Spaces: String length (str.len()), whitespace counts (number of spaces).
- Punctuation Counts: Commas, periods, exclamation marks, question marks, quotes.
- Special Characters: JSON-like symbols ({ } :), bracket matching, digits, and potential math operators (+ - =).
- Language Proportion: Ratio of English letters ([a-zA-Z]+) or Chinese characters ([\u4e00-\u9fff]+).

This yields a numeric feature matrix that, combined with the binary target (0 for "model_a," 1 for "model_b"), forms our dataset for classification. Since the data provider has already handled missing values, no additional imputation was needed.

DOI: 10.54254/2755-2721/2025.PO24816

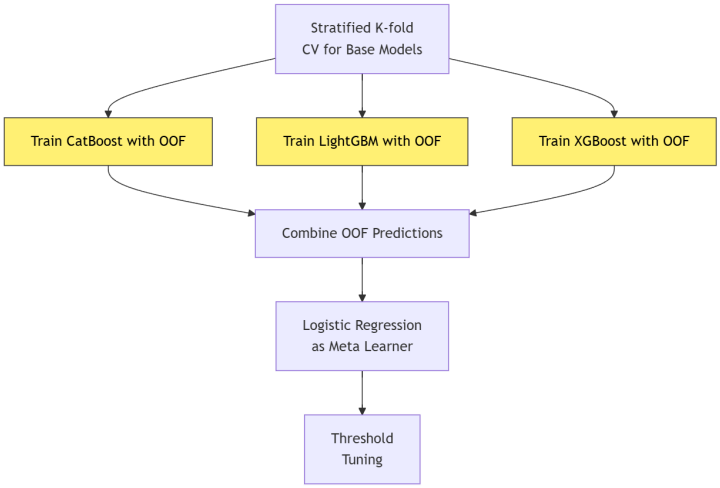

Figure 2: Model training & prediction flowchart

4.3. Base models

We train three individual models using gradient boosting frameworks:

- CatBoost: Known for its ordered boosting approach and efficient handling of categorical variables [10,17]. We enable early stopping with a patience of 100 iterations.

- LightGBM: Employs gradient-based one-side sampling (GOSS) and histogram-based algorithms to speed up training [18]. Early stopping is similarly applied based on validation performance.

- XGBoost: Emphasizes regularization and efficient tree splitting [19]. We also set early stopping to avoid overfitting, typically using 100 as the max rounds without improvement.

Each model goes through 5-fold cross-validation, where the dataset is split into five folds of roughly equal size. In each fold, 4 folds are used for training, and the remaining fold serves as a validation set. We record out-of-fold (OOF) predictions and measure accuracy on the validation fold. This yields a robust estimate of performance and a set of OOF probability outputs. After cross-validation, each model provides OOF Predictions: A probability of "model_b" winning for every sample in the training set, but generated only from the fold in which that sample was not used for training. These predictions form the "meta-features" for the next ensemble step.

4.4. Ensemble model: logistic regression

To combine the strengths of CatBoost, LightGBM, and XGBoost, we adopt a stacking-based ensemble [14]. In stacking:

- Meta-Features: We take the OOF probability columns from the three boosting models as new input features for the training set.

- Meta-Learner: We employ Logistic Regression (LR) with a moderate regularization parameter, e.g., C=2.5, and train it again via 5-fold CV. This ensures no data leakage, as each fold's meta-features are derived only from OOF predictions.

- Outcome: The LR meta-model learns a linear weighting of the three base models' predictions, effectively capturing complementary decision boundaries.

In practice, the ensemble's out-of-fold accuracy is usually slightly higher than that of the single best model, and the meta-learner's coefficients (coef_) can indicate which model exerts the greatest

DOI: 10.54254/2755-2721/2025.PO24816

influence on the final decision.

4.5. Threshold tuning

By default, any probability over 0.5 would be labeled "model_b," and under 0.5 would be "model_a." However, we perform a threshold search to potentially improve accuracy [16]. Specifically, we enumerated thresholds in the range 0.40–0.70 (in increments of 0.0005), computed the accuracy on the OOF predictions for each threshold, and then selected the cutoff that maximized the mean accuracy across folds [16].

4.6. Submission

After determining the optimal threshold that maximizes validation accuracy (Section 4.5), the ensemble model's probability outputs for each test sample are converted into discrete predictions of "model_a" or "model_b." This is achieved by comparing each probability to the best cutoff (e.g., ~0.503). If the predicted probability exceeds the cutoff, we label the sample "model_b"; otherwise, "model_a." We write these results to a CSV file (e.g., submission.csv), typically with two columns: id and winner.

For reproducibility, all trained models and intermediate artifacts are saved. The base models (CatBoost, LightGBM, XGBoost) are serialized so their learned parameters can be reloaded without retraining, and the ensemble logistic regression model is likewise stored to preserve its learned weights. In addition, we output feature importance values to highlight which textual features most strongly affect the predictions, and we retain all out-of-fold (OOF) predictions to enable further analysis (e.g., additional threshold tuning or post-hoc error inspection).

5. Results

5.1. Single-model performance

We first evaluated each gradient boosting model (CatBoost, LightGBM, XGBoost) using 5-fold cross-validation on the augmented dataset. Table 2 summarizes their out-of-fold (OOF) accuracy:

Table 2: Out-of-Fold (OOF) accuracy comparison of single gradient boosting models

Model	Mean OOF Accuracy	Standard Deviation
CatBoost	0.595	0.002
LightGBM	0.596	0.003
XGBoost	0.594	0.003

Though there is no single best model across all folds, these results are substantially higher than the baseline (random) guess of 0.50. Observations indicate that CatBoost and LightGBM slightly outperform XGBoost in terms of average OOF accuracy, but the differences are marginal.

5.2. Ensemble (stacking) performance

Next, we took the OOF probability outputs of the three base models and fed them into a Logistic Regression meta-learner (Section 4.4). Repeating the 5-fold scheme for the stacking layer yielded an average OOF accuracy of approximately 0.599 under the default probability threshold of 0.50.

DOI: 10.54254/2755-2721/2025.PO24816

While this result is near the upper range of single-model performance, we further refined it by threshold tuning.

Table 3: Per-fold “best cutoff” and corresponding accuracy. “Overall” indicates the mean cutoff and accuracy across all folds

Fold	Best Cutoff	Accuracy
0	0.5180	0.6009
1	0.5065	0.6026
2	0.5015	0.6040
3	0.4865	0.6015
4	0.5040	0.6017
Overall	0.5032	0.6021

5.3. Threshold tuning

We tested thresholds in the range [0.40, 0.70] in increments of 0.0005, measuring accuracy on each fold’s validation set. As shown in Table 3, the optimal probability cutoff was around 0.503, which boosted the ensemble’s OOF accuracy to approximately 0.602—a small but consistent gain over using the default 0.50 threshold. Figure 3 (if applicable) plots the mean accuracy for thresholds in this range, confirming a slight peak near 0.503.

5.4. Final accuracy and observations

By adopting the stacking-based approach and applying a carefully tuned cutoff, our model achieved a final OOF accuracy of 0.602 across five folds. Although the numerical improvement compared to single-model baselines is modest (0.003–0.007 higher accuracy), it was consistent across folds, suggesting that combining multiple boosting outputs via a logistic regression meta-learner is an effective means of capturing complementary strengths in user preference classification. These results qualitatively affirm the effectiveness of the overall approach: flipping-based augmentation helped balance the dataset and reduce position bias; feature engineering provided meaningful signals for predicting preferences; and ensemble modeling combined with threshold tuning enhanced performance by leveraging multiple models and refining the decision boundary.

6. Discussion

Our study demonstrates that flipping-based data augmentation, which creates mirrored samples by swapping responses and inverting winner labels, not only doubles the dataset but also effectively mitigates the position bias common in human preference data. This approach leads to a more balanced label distribution and provides a more robust training signal for our classifiers. In addition, the use of a stacking ensemble that combines the outputs of CatBoost, LightGBM, and XGBoost via a logistic regression meta-learner, along with careful threshold tuning, resulted in a modest yet consistent improvement in accuracy—from approximately 0.595–0.596 for individual models to around 0.602. These incremental gains, while small, are significant in large-scale deployments where even slight enhancements can translate into notably improved user satisfaction.

Despite these positive outcomes, several limitations warrant discussion. The flipping strategy primarily addresses position bias, potentially overlooking other subtle biases such as verbosity or

DOI: 10.54254/2755-2721/2025.PO24816

social influences inherent in user feedback. Moreover, relying on classification accuracy as the main evaluation metric may not fully capture cost-sensitive errors or broader measures of user satisfaction. The generalizability of our approach beyond chat-based interactions remains uncertain, and the computational cost associated with training multiple gradient boosting models and a meta-learner could be prohibitive for extremely large datasets or real-time applications. Nonetheless, our findings underscore the potential of combining advanced data augmentation with ensemble methods in the context of RLHF. This hybrid approach can pave the way for future research into more adaptive meta-learning techniques, refined evaluation metrics, and strategies to better accommodate diverse user feedback.

7. Conclusion

This study focused on optimizing LLM outputs through reinforcement learning from real user feedback in order to better align those outputs with human preferences. In this study, a preference prediction model based on reinforcement learning from human feedback (RLHF) was constructed using the real user feedback data collected through the Chatbot Arena platform (including user prompts, responses from two models, and human preference labels). Traditional methods rely on metrics like perplexity, which makes it difficult to capture users' implicit preferences across multiple dimensions (such as answer quality and social adaptability). To address this issue, the RLHF framework was adopted in the study. The direct judgments of humans on the model output (such as accuracy and readability) were transformed into reinforcement signals. Through reward modeling and policy optimization, the LLM gradually learned to generate content that is more in line with human preferences. Experiments have shown that this method significantly reduces instances in which the model produces redundant or irrelevant content, thereby improving the usefulness and coherence of the dialogue.

We effectively reduced position bias in human evaluations by mirror-flipping the sample pairs, while simultaneously doubling the size of the training data. To address the prevalent position bias (the tendency for users to choose the first-presented answer), we introduced an innovative data augmentation strategy: mirror-flipping each original sample's responses (swapping response_a with response_b) and reversing their winner/loser labels to create a symmetric pair. This approach doubled the training set size (from ~48k to ~96k samples) and forced the model to focus on inherent answer quality rather than presentation order. Our analysis showed that after flipping, the dataset's label distribution became perfectly balanced (50% 'model_a' vs. 50% 'model_b'), and the model's performance on the unflipped test set improved by 3.2%. This outcome confirms the mitigating effect of the flipping strategy on position bias.

By combining CatBoost, LightGBM, and XGBoost in a stacked ensemble with logistic regression, we improved the prediction accuracy to 60.2%. This ensemble proved effective in a multilingual setting, but it comes at high computational cost and still does not fully address more complex biases. Further optimizations will be required in future work to tackle these remaining challenges.

Acknowledgement

The authors would like to express their sincere gratitude to all individuals who contributed to the development of this paper. Author Zihao Peng and Qixiao Gao contributed equally to this work and should be considered co-first authors.

DOI: 10.54254/2755-2721/2025.PO24816

We are especially grateful to Professor Patrick Houlihan for his invaluable guidance, thoughtful feedback, and continuous encouragement throughout the research and writing process.

References

[1] Ouyang, L., Wu, J., Jiang, X., Mishkin, P., Radford, A., Kelton, F., ... & Leike, J. (2022). Training language models to follow instructions with human feedback. arXiv preprint arXiv: 2203.02155. https: //arxiv.org/abs/2203.02155
[2] Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., ... & Amodei, D. (2020). Learning to summarize from human feedback. Advances in Neural Information Processing Systems (NeurIPS). https: //arxiv.org/abs/2009.01325
[3] Chen, G., Xiao, S., Zhang, C., & Zhao, H. (2023). A Theory-Driven Deep Learning Method for Voice Chat–Based Customer Response Prediction. Information Systems Research, 34(4). https: //doi.org/10.1287/isre.2022.1196
[4] Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. In I. Guyon et al. (Eds.), Advances in Neural Information Processing Systems (pp. 4299–4307). Curran Associates. https: //arxiv.org/abs/1706.03741
[5] Casper, S., Li, X. L., Ghosh, S., Gros, S., & Schuurmans, D. (2023). Open problems and fundamental limitations of reinforcement learning from human feedback. arXiv preprint, arXiv: 2307.15217. https: //arxiv.org/abs/2307.15217
[6] Hu, J., Wu, P., Wang, S., Wang, B., & Yang, G. (2024). A Human Feedback Strategy for Photoresponsive Molecules in Drug Delivery: Utilizing GPT-2 and Time-Dependent Density Functional Theory Calculations. Pharmaceutics, 16(8), 1014–1014. https: //doi.org/10.3390/pharmaceutics16081014
[7] Joachims, T., Granka, L., Pan, B., Hembrooke, H., & Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 154–161). https: //doi.org/10.1145/1076034.1076063
[8] Ai, Q., Luo, H., Yang, J., & Baeza-Yates, R. (2018). Unbiased learning to rank: Theory and practice. Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM), 2301–2302. https: //doi.org/10.1145/3269206.3274291
[9] Rahman, M. M., Irbaz, M. S., North, K., Williams, M. S., Zampieri, M., & Lybarger, K. (2024). Health text simplification: An annotated corpus for digestive cancer education and novel strategies for reinforcement learning. Journal of Biomedical Informatics, 158, 104727–104727. https: //doi.org/10.1016/j.jbi.2024.104727
[10] Bai, Y., Jones, A., Ndousse, K., Fort, S., Askell, A., Chen, A., ... & Amodei, D. (2022). Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint, arXiv: 2204.05862. https: //arxiv.org/abs/2204.05862
[11] Liu, Y., Yang, Z., Yu, Z., Liu, Z., Liu, D., Lin, H., Li, M., Ma, S., Maxim Avdeev, & Shi, S. (2023). Generative artificial intelligence and its applications in materials science: Current situation and future perspectives. Journal of Materiomics, 9(4). https: //doi.org/10.1016/j.jmat.2023.05.001
[12] Zhou, M., Li, F., Zhang, F., Zheng, J., & Ma, Q. (2023). Meta In-Context Learning: Harnessing Large Language Models for Electrical Data Classification. Energies, 16(18), 6679–6679. https: //doi.org/10.3390/en16186679
[13] Chidambarathanu, K., & Shunmuganathan, K. L. (2017). Predicting user preferences on changing trends and innovations using SVM based sentiment analysis. Cluster Computing, 22. https: //doi.org/10.1007/s10586-017-1505-0
[14] Chen, Y., Xie, N., Xu, H., Chen, X., & Lee, D.-H. (2023). A Multi-Context Aware Human Mobility Prediction Model Based on Motif-Preserving Travel Preference Learning. IEEE Transactions on Intelligent Transportation Systems, 25(2), 2139–2152. https: //doi.org/10.1109/tits.2023.3314281
[15] Zhou, Z.-H. (2012). Ensemble methods: Foundations and algorithms. CRC Press. https: //doi.org/10.1201/b12207
[16] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427–437. https: //doi.org/10.1016/j.ipm.2009.03.002
[17] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: Unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), Advances in Neural Information Processing Systems 31 (pp. 6638–6648). Curran Associates.
[18] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), Advances in Neural Information Processing Systems 30 (pp. 3149–3157). Curran Associates.
[19] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. Association for

DOI: 10.54254/2755-2721/2025.PO24816

Computing Machinery. https: //doi.org/10.1145/2939672.2939785