Lightweight and Scalable Model for Tweet Engagements Predictions in a Resource-constrained Environment

Luca Carminati
Politecnico di Milano
Milan, Italy
luca5.carminati@mail.polimi.it

Giacomo Lodigiani
Politecnico di Milano
Milan, Italy
giacomo.lodigiani@mail.polimi.it

Pietro Maldini
Politecnico di Milano
Milan, Italy
pietro.maldini@mail.polimi.it

Samuele Meta
Politecnico di Milano
Milan, Italy
samuele.meta@mail.polimi.it

Stiven Metaj
Politecnico di Milano
Milan, Italy
stiven.metaj@mail.polimi.it

Arcangelo Pisa
Politecnico di Milano
Milan, Italy
arcangelo.pisa@mail.polimi.it

Alessandro Sanvito
Politecnico di Milano
Milan, Italy
alessandro1.sanvito@mail.polimi.it

Mattia Surricchio
Politecnico di Milano
Milan, Italy
mattia.surricchio@mail.polimi.it

Fernando B. Pérez Maurera
Politecnico di Milano
Milan, Italy
fernandobenjamin.perez@polimi.it

Cesare Bernardis
Politecnico di Milano
Milan, Italy
cesare.bernardis@polimi.it

Maurizio Ferrari Dacrema
Politecnico di Milano
Milan, Italy
maurizio.ferrari@polimi.it

ABSTRACT

In this paper we provide an overview of the approach we used as team Trial&Error for the ACM RecSys Challenge 2021. The competition, organized by Twitter, addresses the problem of predicting different categories of user engagements (Like, Reply, Retweet and Retweet with Comment), given a dataset of previous interactions on the Twitter platform. Our proposed method relies on efficiently leveraging the massive amount of data, crafting a wide variety of features and designing a lightweight solution. This results in a significant reduction of computational resources requirements, both during the training and inference phase. The final model, an optimized LightGBM, allowed our team to reach the 4th position in the final leaderboard and to rank 1st among the academic teams.

CCS CONCEPTS

• Information systems → Recommender systems; **• Computing methodologies → Classification and regression trees**; **Neural networks**.

KEYWORDS

ACM RecSys Challenge 2021, Recommender Systems, Gradient Boosting for Decision Trees, Neural Networks

RecSysChallenge 2021, October 1, 2021, Amsterdam, Netherlands
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8693-7/21/10…$15.00
https://doi.org/10.1145/3487572.3487597

ACM Reference Format:
Luca Carminati, Giacomo Lodigiani, Pietro Maldini, Samuele Meta, Stiven Metaj, Arcangelo Pisa, Alessandro Sanvito, Mattia Surricchio, Fernando B. Pérez Maurera, Cesare Bernardis, and Maurizio Ferrari Dacrema. 2021. Lightweight and Scalable Model for Tweet Engagements Predictions in a Resource-constrained Environment. In *RecSysChallenge '21: Proceedings of the Recommender Systems Challenge 2021 (RecSysChallenge 2021), October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3487572.3487597

1 INTRODUCTION

Recommender Systems have established themselves as a useful tool to offer personalized and relevant content in many different sectors. Social networks are no exception, since accurate recommendations can substantially impact the user experience. The ACM RecSys Challenge 2021 [2], organized by Twitter, aims at identifying the best approach to predict different user engagements (Like, Reply, Retweet, Retweet with comment) with a certain tweet. This prediction task comes with several other challenges: the need to handle a dataset at scale (approximately $1B$ data points), the limited computational resources available for the evaluation and the inclusion of *fairness* as an evaluation metric. Hence, solutions must be scalable and maintain high accuracy for different categories of users. We propose an optimized LightGBM model, which leverages a wide variety of meaningful features. The proposed solution strikes a balance between highly accurate predictions, scalability and fairness, allowing our team to reach the 4th position with an inference time from two to three times lower than the other top participants. The source code of our final model and the respective documentation are publicly available on Github.[1]

[1] https://github.com/recsyspolimi/recsys-challenge-2021-twitter

The paper is organized as follows. In Section 2 we introduce the problem, the dataset and the evaluation metrics of the Challenge. In Section 3 we discuss the data splitting for validation and feature extraction. In Section 4 we present in detail the feature engineering process and some important features. In Section 5 we describe the tested models and their hyperparameter optimization. In Section 6 we report the experimental results of the most promising models. Lastly, in Section 7 we draw the conclusions.

2 PROBLEM FORMULATION

The ACM RecSys Challenge 2021 required participants to predict the probability that a user will interact (*engage*) with a tweet in a certain way. In particular 4 kinds of engagements are possible: Like, Reply, Retweet, Retweet with comment (we refer to this class as Comment in the text). Although the goal of predicting user engagements is similar to that of the ACM RecSys Challenge 2020 [4], this Challenge included both resource constraints and fairness metrics in the evaluation. These constraints are set to encourage teams to propose novel solutions.

Dataset [5]. For the purpose of this Challenge, Twitter released a dataset of approximately $1B$ data points spanning over a period of 28 days. Each data point represents a *possible* engagement of a user with a tweet. Engagements that occurred are referred to as *positive samples*, while engagements that did not occur are referred to as *negative samples*. The first three weeks are used as training data, while the data points in the last week are randomly split in half for the validation and test data. The ground truth of the validation data points was released in the last two weeks of the Challenge. Tweets removed from the platform also had to be removed from the dataset to ensure compliance with privacy laws. Thus, several versions of the dataset were released during the Challenge. At the end of the Challenge the training and validation data contained $629.5M$ and $14.5M$ data points respectively. The data points have features associated with them, such as the content of the tweet represented as BERT tokens [9], the account of the creator of the tweet, the account of the user which may engage with the tweet, and a timestamp associated with the engagement if it occurred, i.e., if it is a positive sample. The dataset is strongly unbalanced towards negative samples. The percentage of positive samples is: 39.38% for Like, 8.63% for Retweet, 2.65% for Reply, 0.66% for Comment.

Computational resources. The predictions had to be computed on a cloud computing instance provided by Twitter. This instance was constrained to have only one CPU, 64GB RAM, and 24 hours of total inference time. Solutions exceeding the 24 hours limit were rejected.

Multi-objective evaluation. The Challenge rewards both accurate and *fair* predictions, penalizing recommendations that are biased towards more popular tweet creators. To account for fairness, authors of tweets are categorized into 5 groups, according to the quantiles of their follower counts. The accuracy of each group is computed as the Average Precision (AP) and the Relative Cross-Entropy (RCE) considering only the tweets authored by a user in that group. The final AP and RCE scores for each engagement are obtained as the average of the 5 group scores. Finally, submissions are first ranked by RCE and AP separately, taking the average of the considered metric for the 4 engagements; then, the ranking for the 2 metrics are summed to obtain the Overall Score shown on the leaderboard.

3 DATA SPLIT

The dataset released for the final phase of the Challenge consists of two parts: a training set containing $629.5M$ interactions and a validation set containing $14.5M$ interactions. For our experiments, we split the available data in three non-overlapping subsets (see Figure 1):

- The first subset is used to extract and compute the features. It is composed of the union between the 80% of the data in the original training set and the 70% of the data in the validation.
- The second subset is used to train our models. It is composed of the 20% of the validation and the most recent engagements accounting for the 20% of the data points in the training set not included in the first subset.
- The third subset is used to validate our models. It is composed of the remaining 10% of validation data not included in the previous subsets.

During the split operation it is of great importance to find a balance between data coming from the original training and validation sets. Indeed, data available in the validation set is particularly valuable, due to the similarity between validation and test sets discussed in Section 2 (i.e., data points in validation and test sets were gathered during the same week). For the same reason, we applied a temporal filtering on the training data points that compose the second subset used for model training, keeping only the most recent 20% as previously mentioned.

Since the datasets contained both training and validation data, we created a boolean feature to indicate whether any interaction came from the original train or validation sets. Our experiments indicate that including this feature into models improves the accuracy of predictions.

4 FEATURES

Feature engineering proved crucial to improve the accuracy of our model. The similarity with the ACM RecSys Challenge 2020 allowed us to benefit from the experience of the best teams. We analyzed the list of features presented in [11, 16] and we selected the most relevant ones, according to the respective authors. However, due to the characteristics of the Challenge, we had to perform implementation-side modifications to integrate the features in our new solution and to satisfy the tighter constraints on the computation. In particular, because of the size of the dataset, we leveraged the capabilities of highly scalable libraries (such as Dask[7]). Indeed, preprocessing and feature extraction had to be efficient both in CPU and memory usage, due to the low amount of resources available on the computing instance provided by Twitter at inference time. Finally, other features were added to cope with the required focus on fair recommendations of the evaluation metrics. In the following sections we will describe some of the most important features we used.

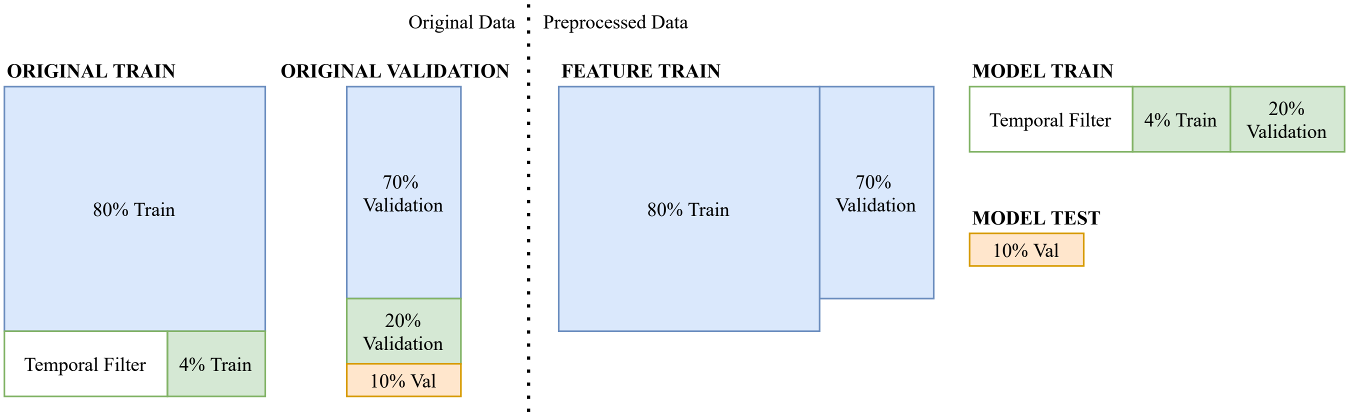

Figure 1: Data splits used during the challenge. Blue splits are used for feature engineering, green splits for models' training, and orange splits for models' testing.

4.1 Text-based Features

The text of each tweet, provided as BERT tokens, was used to generate counting features, with the purpose of representing both the syntactic structure and semantic content of a tweet in a form better suited for the model. To characterize the syntactic structure, specific counts on the tokenized tweet were performed to count different types of punctuation symbols or the number of tokens in the tweet. To characterize the semantic structure, each tweet was detokenized and a counting function over specific sets of thematic words in different languages was applied. This allowed to characterize the tweet content in relation to various trending topics such as *covid19*, *sports*, *news*, *videogames*. We produced those lists of thematic words by integrating heterogeneous sources. Most of the words were identified through a manual procedure, analyzing the detokenized text of the tweets containing the most popular hashtags. Additional topics were selected by analyzing the Google Trends for the period of our data.

4.2 Memory-based Features

Memory-based features reflect the characteristics of the dataset by analyzing the available data. We employed them to effectively associate each interaction to broader-view information regarding past trends of users and tweets.

4.2.1 Counts. These features consist in counting each possible combination of two categorical features, eventually filtering the count according to a required type of interaction. This type of feature allows having a quantitative information of the past co-occurrences of specific values for pairs of features. An example is to count the occurrences of each tuple (userID, language), to identify the number of past Like/Comment/Retweet/Reply/All interactions that involved a specific pair of user and language. This type of information helped the model to weight differently the predictions, depending on the past history of the user (see Section 6). To address the difficulties of counting over possibly high-cardinality features in a distributed environment, we implemented the counting procedures taking advantage of sparse matrices and ad-hoc reductions to sum the counts computed concurrently over different partitions of the dataset.

4.2.2 Target Encoding. In the released dataset, a significant portion of the available information on the users was represented by categorical features, mostly by userIDs. Since the cardinality of such features ranged in the tens of millions, representing them as one-hot encoded features was not feasible. Therefore, we used the statistics of the 4 engagements (*targets*) for the encoding of the categorical values, as proposed in [14]. Given a list of one or more feature values, we calculated the frequency of those feature values appearing together with the target label. To account for high variance of statistics of values appearing just a few times in the dataset, we adopted the smoothing strategy proposed in [16]. Target encoding gave us a great boost in terms of prediction accuracy, but it required a large amount of data for reliable statistics. Hence, in our final approach we divided the feature engineering dataset in several parts, we calculated the target encoding for each part, and we finally averaged them. This approach proved more efficient, allowing to obtain the same accuracy with lower resource requirements.

5 MODELS

Due to the considerable size of the dataset, the models used for predictions had to be scalable and fast. We experimented with three different models, selected based on the results presented in the previous ACM RecSys Challenge [11, 12, 16].

5.1 Neural Networks

Neural Network (NN) models are a widely researched topic that had strong impact on current research literature, but their effectiveness in competitions is varied, being sometimes the winning strategy [8] and sometimes not [12]. In our experiments we tested a NN model trained on all the available features (see Section 4). To avoid overfitting and increase the generalization capability of the model, we adopted a simple architecture composed of 3 hidden layers with 256, 128 and 64 neurons, respectively. We included dropout, batch normalization and input normalization strategies. We trained four

Table 1: Metrics with fairness measured on the subset we use for local validation described in Section 3.

	AP				RCE			
Model	**Retweet**	**Reply**	**Like**	**Comment**	**Retweet**	**Reply**	**Like**	**Comment**
XGBoost	0.4034	**0.2070**	0.6629	0.0690	23.6291	**21.7480**	18.0909	13.8495
LightGBM	**0.4056**	0.2064	**0.6685**	**0.0707**	**23.8161**	21.719	**18.4962**	**14.1673**
NN	0.3852	0.2001	0.6614	0.0598	22.4358	21.2941	17.2435	12.7172

Table 2: Top 5 submissions in the final leaderboard of the Challenge.

Rank	Team	Method	Inference Time	Overall Score
1	NVIDIA	nvidia_rapidsai_final_ensemble_v2	23 hours	2
2	SYNERISE	Synerise_v1	18 hours	4
3	LAYER 6	LAYER6_AI	13 hours	6
4	**Trial&Error**	**test_lightgbm**	**5 hours**	**8**
5	perecasxiru	final1	19 hours	10

different NN, one for each type of engagement, sharing the same hyperparameters.

5.2 Gradient Boosting for Decision Trees

Gradient Boosting for Decision Tree (GBDT) models are the state-of-the-art solution for very sparse datasets. Similar to NNs, we trained four different GBDT models, one for each type of engagement, using as input all the features we developed. In our experiments we tested two of the most popular implementations of GBDT models: XGBoost [6] and LightGBM [13]. We adopted LightGBM as it provided the highest prediction accuracy with lower resource requirements.

5.3 Hyperparameter tuning

The optimization of the hyperparameters is a very important step to maximize the effectiveness of the predictive models [3, 11]. We adopted different hyperparameter tuning procedures depending on the model. For NN, we performed a random search followed by a manual fine-tuning. For LightGBM, we performed Bayesian Optimization using Optuna [1]. More specifically, we adopted the "Stepwise algorithm" that tunes important hyperparameters sequentially, resulting in a compact search space [15]. We decided to use the binary log-loss as evaluation metric for the optimization, since it was strongly correlated with both RCE and AP and required significantly less time for its computation.

6 RESULTS

In this section we provide an overview and a discussion of our results along the three relevant dimensions of accuracy, performance and fairness. All the results shown in this section are obtained using the data splits as described in Section 3.

Prediction accuracy. In Table 1 we show the accuracy of the models presented in Section 5. LightGBM outperformed XGBoost and the NN in almost all classes for AP and RCE, except for Reply, where XGBoost had higher AP and RCE. Although the NN was the least accurate model, its accuracy was *close* to both XGBoost and LightGBM even with minimal tuning. However, due to its very high computational cost it was unfeasible to thoroughly optimize its hyperparameters within the resource constraints and short duration of the Challenge, which is likely a factor in its lower accuracy. Due to its consistently superior accuracy, LightGBM was our choice for the final submission.

Performance. The choice of LightGBM was also driven by the faster training and prediction times we observed during the Challenge compared to the other models. In Table 2 we show that, thanks to the choice of the model and the efficient preprocessing described in Section 4, our solution has the lowest inference time among the top 5 of the official leaderboard, taking only 5 hours compared to the 13 hours required by the second-fastest and the 23 hours required by the slowest (but most accurate) method.

A considerable speedup in training time was achieved by leveraging training samples temporally close to the ones in the test set, as described in Section 3. This resulted into both a substantial reduction in the number of samples used to train the model, and an improvement in the accuracy of the predictions. Indeed, this solution scored 2% AP and 3% RCE more than an instance of LightGBM trained on twice the amount of data points, randomly selected from training and validation sets described in Section 3[2].

Fairness. In Figure 2 we show the difference in accuracy across the different quantiles for LightGBM (i.e., the model used in our final submission), which exhibited remarkably consistent results. The most relevant exception is represented by a sudden increase of AP measurement over the first quantile in the Comment class. This quantile scores 0.1511 in AP, an outstanding 113.75% more than the average AP for this engagement class. This behavior is likely due to the lack of support of the Comment class in the first quantile. Data points from this quantile account for 0.15% of the total data points in the dataset we use for validation, where only 1.02% of them (30 in absolute terms) represent positive interactions in the Comment class. The RCE of the first quantile in the same class exhibits a similar pattern, being 18.60% higher than the overall RCE. The RCE of the first quantile in the Retweet class represents another exception, although it exhibits an opposite pattern compared to the Comment class. This quantile scores 17.3358 in RCE, which is

[2] The dataset used for training was composed of 10% of the training set and 10% of validation set selected uniformly at random.

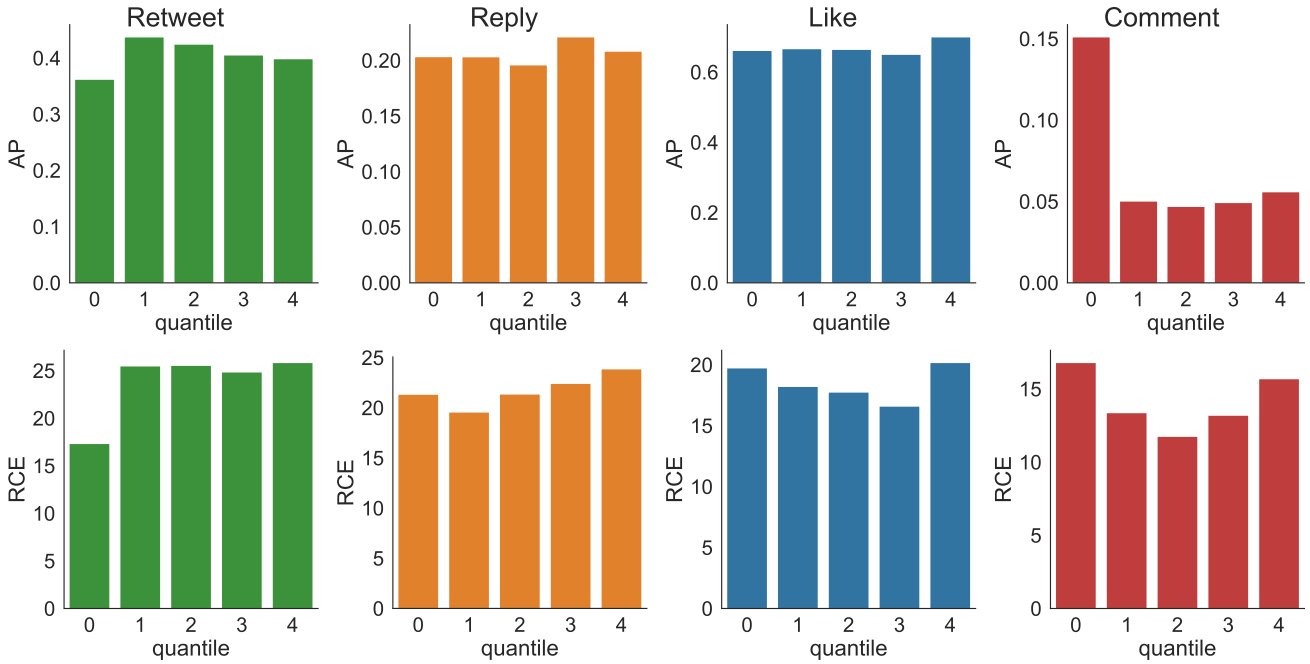

Figure 2: LightGBM's metrics by quantile and engagement measured on our local test data, described in Section 3.

27.21% less than the average RCE for this engagement class. This time, support is higher, with 10.71% of the data points from the first quantile representing poisitive interactions in the Retweet class. Overall, it is interesting to notice that there is not a clear correlation between the quantile (i.e., the popularity of the author of a tweet) and the accuracy of the model.

Feature importance. To assess the relevance of each feature in the model used in our final solution and gain insights about what the model learned, we computed the permutation importance using Eli5 [10]. The most important features are the target encodings of "engager ID" and "creator ID" with respect to the target class. They represent, respectively, the likelihood that the given user interacts with a tweet and the likelihood that an engagement occurs on a tweet of a given author. For all classes, target encoding of multiple features (the combination of domains, tweet language, tweet type, engager follows creator, the number of photos and creator is verified) represents a valuable feature. Specifically for Like and Retweet classes, media and language content features are particularly important (i.e., the presence of photos or videos in the tweet, the presence of a language with which the user had previous positive interactions). For the Reply and Comment (to a lesser extent) classes, the number of question marks plays an important role in the prediction. This suggests that tweets with an open question are more likely to be answered by users.

7 CONCLUSIONS

The ACM RecSys Challenge 2021 aimed at predicting, given a tweet, the probability of a user's engagement. Starting from the original dataset, we crafted a large set of features relying on the text of the tweet, target encoding and counting operations. This allowed us to train the final model, a set of four optimized LightGBMs, one for each type of engagement, with a more comprehensive representation of the problem. Moreover, the peculiar dataset splitting strategy and the effective feature extraction process paved the way for a fast and accurate approach, whose inference time was two to three times lower than solutions proposed by other top participants. As final result, our team Trial&Error placed first among the academic teams and ranked 4th in the final leaderboard.

ACKNOWLEDGMENTS

We would like to thank Prof. Paolo Cremonesi for his support.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 2623–2631. https://doi.org/10.1145/3292500.3330701

[2] Vito Walter Anelli, Saikishore Kalloori, Bruce Ferwerda, Luca Belli, Alykhan Tejani, Frank Portman, Alexandre Lung-Yut-Fong, Ben Chamberlain, Yuanpu Xie, Jonathan Hunt, Michael M. Bronstein, and Wenzhe Shi. 2021. RecSys 2021 Challenge Workshop: Fairness-aware engagement prediction at scale on Twitter's Home Timeline. In *RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September 2021 - 1 October 2021*, Humberto Jesús Corona Pampín, Martha A. Larson, Martijn C. Willemsen, Joseph A. Konstan, Julian J. McAuley, Jean Garcia-Gathright, Bouke Huurnink, and Even Oldridge (Eds.). ACM, 819–824. https://doi.org/10.1145/3460231.3478515

[3] Sebastiano Antenucci, Simone Boglio, Emanuele Chioso, Ervin Dervishaj, Shuwen Kang, Tommaso Scarlatti, and Maurizio Ferrari Dacrema. 2018. Artist-driven layering and user's behaviour impact on recommendations in a playlist continuation scenario. In *Proceedings of the ACM Recommender Systems Challenge, RecSys Challenge 2018, Vancouver, BC, Canada, October 2, 2018.* ACM, 4:1–4:6. https://doi.org/10.1145/3267471.3267475

[4] Luca Belli, Sofia Ira Ktena, Alykhan Tejani, Alexandre Lung-Yut-Fong, Frank Portman, Xiao Zhu, Yuanpu Xie, Akshay Gupta, Michael M. Bronstein, Amra

Delic, Gabriele Sottocornola, Vito Walter Anelli, Nazareno Andrade, Jessie Smith, and Wenzhe Shi. 2020. Privacy-Preserving Recommender Systems Challenge on Twitter's Home Timeline. *CoRR* abs/2004.13715 (2020). arXiv:2004.13715 https://arxiv.org/abs/2004.13715

[5] Luca Belli, Alykhan Tejani, Frank Portman, Alexandre Lung-Yut-Fong, Ben Chamberlain, Yuanpu Xie, Kristian Lum, Jonathan Hunt, Michael Bronstein, Vito Walter Anelli, Saikishore Kalloori, Bruce Ferwerda, and Wenzhe Shi. 2021. The 2021 RecSys Challenge Dataset: Fairness is not optional. arXiv:2109.08245 [cs.SI]

[6] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 785–794. https://doi.org/10.1145/2939672.2939785

[7] Dask Development Team. 2016. *Dask: Library for dynamic task scheduling.* https://dask.org

[8] Gabriel de Souza Pereira Moreira, Sara Rabhi, Ronay Ak, Md Yasin Kabir, and Even Oldridge. 2021. Transformers with multi-modal features and post-fusion context for e-commerce session-based recommendation. *CoRR* abs/2107.05124 (2021). arXiv:2107.05124 https://arxiv.org/abs/2107.05124

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[10] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long Form Question Answering. *CoRR* abs/1907.09190 (2019). arXiv:1907.09190 http://arxiv.org/abs/1907.09190

[11] Nicolò Felicioni, Andrea Donati, Luca Conterio, Luca Bartoccioni, Davide Yi Xian Hu, Cesare Bernardis, and Maurizio Ferrari Dacrema. 2020. Multi-Objective Blended Ensemble For Highly Imbalanced Sequence Aware Tweet Engagement Prediction. In *RecSys Challenge '20: Proceedings of the Recommender Systems Challenge 2020, Virtual Event Brazil, September, 2020.* ACM, 29–33. https://dl.acm.org/doi/10.1145/3415959.3415998

[12] Dietmar Jannach, Gabriel de Souza Pereira Moreira, and Even Oldridge. 2020. Why Are Deep Learning Models Not Consistently Winning Recommender Systems Competitions Yet?: A Position Paper. In *RecSys Challenge '20: Proceedings of the Recommender Systems Challenge 2020, Virtual Event Brazil, September, 2020.* ACM, 44–49. https://dl.acm.org/doi/10.1145/3415959.3416001

[13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 3146–3154. https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[14] Daniele Micci-Barreca. 2001. A Preprocessing Scheme for High-Cardinality Categorical Attributes in Classification and Prediction Problems. *SIGKDD Explor.* 3, 1 (2001), 27–32. https://doi.org/10.1145/507533.507538

[15] Jean-François Puget. 2019. Beyond Feature Engineering and HPO. https://www.youtube.com/watch?v=VC8Jc9_lNoY

[16] Benedikt Schifferer, Gilberto Titericz, Chris Deotte, Christof Henkel, Kazuki Onodera, Jiwei Liu, Bojan Tunguz, Even Oldridge, Gabriel de Souza Pereira Moreira, and Ahmet Erdem. 2020. GPU Accelerated Feature Engineering and Training for Recommender Systems. In *RecSys Challenge '20: Proceedings of the Recommender Systems Challenge 2020, Virtual Event Brazil, September, 2020.* ACM, 16–23. https://dl.acm.org/doi/10.1145/3415959.3415996