RecSys 2021 Challenge Workshop: Fairness-aware engagement prediction at scale on Twitter's Home Timeline

VITO WALTER ANELLI, Politecnico di Bari, Italy
SAIKISHORE KALLOORI, ETH Zürich, Switzerland
BRUCE FERWERDA, Jönköping University, Sweden
LUCA BELLI, Twitter Inc., USA
ALYKHAN TEJANI, Twitter Inc., UK
FRANK PORTMAN*, Twitter Inc., USA
ALEXANDRE LUNG-YUT-FONG*, Twitter Inc., UK
BEN CHAMBERLAIN, Twitter Inc., UK
YUANPU XIE, Twitter Inc., USA
JONATHAN HUNT, Twitter Inc., UK
MICHAEL BRONSTEIN, Twitter Inc., UK
WENZHE SHI, Twitter Inc., UK

The workshop features presentations of accepted contributions to the RecSys Challenge 2021, organized by Politecnico di Bari, ETH Zürich, Jönköping University, and the data set is provided by Twitter. The challenge focuses on a real-world task of tweet engagement prediction in a dynamic environment. For 2021, the challenge considers four different engagement types: Likes, Retweet, Quote, and replies. This year's challenge brings the problem even closer to Twitter's real recommender systems by introducing latency constraints. We also increases the data size to encourage novel methods. Also, the data density is increased in terms of the graph where users are considered to be nodes and interactions as edges. The goal is twofold: to predict the probability of different engagement types of a target user for a set of Tweets based on heterogeneous input data while providing fair recommendations. In fact, multi-goal optimization considering accuracy and fairness is particularly challenging. However, we believed that the recommendation community was nowadays mature enough to face the challenge of providing accurate and, at the same time, fair recommendations. To this end, Twitter has released a public dataset of close to 1 billion data points, > 40 million each day over 28 days. Week $1 - 3$ will be used for training and week 4 for evaluation and testing. Each datapoint contains the tweet along with engagement features, user features, and tweet features. A peculiarity of this challenge is related to keeping the dataset updated with the platform: if a user deletes a Tweet, or their data from Twitter, the dataset is promptly updated. Moreover, each change in the dataset implied new evaluations of all submissions and the update of the leaderboard metrics. The challenge was well received with 578 registered users, and 386 submissions.

CCS Concepts: • **Information systems → Recommender systems**.

Additional Key Words and Phrases: Recommender Systems; Fairness; Online Social Networks; Embeddings; BERT

*Both authors contributed equally to this research.

Manuscript submitted to ACM

ACM Reference Format:
Vito Walter Anelli, Saikishore Kalloori, Bruce Ferwerda, Luca Belli, Alykhan Tejani, Frank Portman, Alexandre Lung-Yut-Fong, Ben Chamberlain, Yuanpu Xie, Jonathan Hunt, Michael Bronstein, and Wenzhe Shi. 2021. RecSys 2021 Challenge Workshop: Fairness-aware engagement prediction at scale on Twitter's Home Timeline. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460231.3478515

1 INTRODUCTION

In conjunction with the annual ACM Recommender Systems (RecSyS) conference, a unique recommendation challenge is proposed to academia and industry: the RecSys Challenge. The goal for participants in the challenge is to develop novel approaches and techniques on the data-sets provided. A key aspect of the RecSys Challenge is that the provided data-sets originate from real-world applications. For example, the data for 2017 was provided by Xing on job recommendations [1], 2018 by Spotify on music recommendations [7], 2019 by Trivago on travel recommendations [12], and 2020 by Twitter on content recommendations [5]. The collaboration with Twitter was continued for the challenge of 2021. [1].

An important aspect for Twitter is to display relevant and healthy content (e.g., from breaking news and entertainment to sports, politics and everyday interests) to their users on a local as well as on a global level. Especially with the influx of everyday information, it is vital that content ("tweets") reaches the right users to interact with. Interaction with tweets generally happens through likes, replies, retweets, and quote tweets. For the majority of Twitter users, their timeline is the default starting point to catch up with the latest content. The content in the timeline can either be displayed in reverse chronological order or ranked algorithmically. For the algorithmic ranking each relevant tweet is scored to estimate the degree of interest and engagement for the user. Especially the quality of the algorithmic ranking is vital for the quality of the user experience. Users have more interactions and are more likely to come back to the platform when the timeline shows the most relevant tweets first.

Determining the algorithmic quality is challenging as quality itself is very subjective. For the RecSys Challenge the quality is proxied through the use of engagement. Engagement itself can be measured through content interactions (i.e., likes, replies, retweets). The feedback received from users on the displayed content is only implicit, since users do not explicitly rate, nor rank tweet relevance on their timeline. The lack of explicit feedback makes this an even more difficult task. Moreover, to date, the lack of public datasets for user engagement prediction can be noted, which further signifies the relevance of this year RecSys Challenge. To this end, Twitter has released a large public dataset of ~1 billion samples from Twitter's timeline, split almost equally between positive and negative examples, paying particular attention to user privacy. The published dataset is the largest public dataset ever released by a social network platform, and it can significantly help advance the state-of-the-art in user recommendations with implicit feedback.

2 RECSYS 2021 CHALLENGE

Twitter is a microblogging platform that is mainly used for sharing news and for people to share their thoughts with a big audience through short messages named "tweets." Finding relevant information on Twitter can be done through following users or specific topics through hashtags. Subsequently interaction with a tweet can take place in various ways such as through likes, replies, retweets, and quote tweets. The prior interaction with tweets are important determinants to decide what kind of recent tweets to display on the user's timeline.

The timeline provides the user with an overview of the most recent tweets of interest. Especially when the tweets in the timeline is ranked algorithmically, the importance of the interactions with prior tweets become apparent as a

[1] https://recsys.acm.org/recsys21/challenge/

prediction is made on the relevance of each tweet for the user. The tweets that are ranked as more relevant will be presented higher in the order on the timeline of the user.

Given that tweets are short due the limited use of 280 characters, the content tend to be fast paced in which the tweet conversation can change. Hence, there is an importance in providing the user with the most timely and relevant content to further expand the engagement. For the RecSys 2021 Challenge the following problem statement is posed:

Given a (reader, tweet) pair, what is the probability that the reader will engage with the tweet?

Engagement is defined as one of four different types: *like* (reader clicks the like button), *reply* (reader replies to a tweet, thus creating a new tweet themselves), *retweet* (reader shares the original tweet with their followers) and *quote* (reader adds their comment or media before sharing the original tweet).

2.1 A Challenge at Scale

In this section, some of the common industry scalability challenges that participants of the RecSys 2021 Challenge needed to tackle are described.

Feature Selection: Social Graph While feature selection is a common problem for machine learning (ML) practitioners, the social graph problem is focused on in specific. On Twitter, users can follow each other. Since the number of unique users is on the order of hundreds of millions, this size poses the problem of how to practically encode this follower-followee relationship. If each users is one-hot encoded the feature space would explode in dimension. This design choice will cause all sorts of engineering problems: more data is required to avoid overfitting, more hardware is required, and latency might increase (also see Section 2.1).

Sampling and label imbalance Each day hundreds of millions of tweets are produced by users all across the world. Ideally, when training a ML model is to use all available data (after splitting in train, test, and validation). However, reality learns that data of such a significant size requires extensive time and resources to train the model. Hence, a representative sample is used to find a balance between dataset size and the needed time and resources to process. In the case of Twitter, there is also a big label imbalance between positive and negative examples. A positive example is a tweet that the user saw and engaged with, while a negative one is a tweet that the user saw but decided *not* to engage with. Users tend to engage with a fraction of the tweets they see. This behavior translates into a strong imbalance between the positive and negative classes.

Latency The real-time nature of Twitter makes latency a primary concern. When a user logs in on the platform and requests a new timeline (i.e., a fresh set of tweets to see on their timeline), it is important that the ranking is done in real time and must hold true for all the hundreds of millions of users that refresh their timeline every day. Any model that requires more than a few millisecond to score would be considered too slow and not suitable for production settings. In the RecSys 2021 Challenge the imposed production constraints were avoided. At the same time participants were nudged away from slower and more complex models (e.g., ensemble ones) even at the cost of accuracy in some cases. While during the training stage, participants were free to train on hardware of their choosing, for the testing phase, participants were asked to upload their code to run on a dedicated Docker instance (1 CPU with 64GB of RAM). Furthermore, a 24h timeout limit was imposed for the entirety of the dataset.

Intrinsic value of metrics Ideally, the ranking of the tweets would reflect the readers' preferences, with the most valuable ones being on top. In practice, it is hard to asses the intrinsic value of a piece of content, since users are not asked to rank or score tweets. This in contrast to movie streaming service where a user can rank/rate each piece of content. While engagement is the industry-standard metric (i.e., a user found a tweet valuable when engaging with a tweet), it is important to mention that it is still a proxy-metric, and that in the current settings it is hard to disentangle real value from engagements (e.g., [11, 13]).

Fairness A new addition to the RecSys Challenge is the addition of fairness considerations. Fairness is a societal concept, not an optimization one [17] and it can come in many different shape or forms [8]. When defining what fairness means for the purpose of this challenge, three important aspects were considered. First, we introduced the fairness space to the participants of the challenge without requiring too much prior knowledge or special infrastructure. Second, it was necessary to define a fairness space relevant for Twitter and consequently for the challenge. Finally, the sensitive attributes should already be available in the dataset (see Section 3.1). Therefore, demographic parity metrics would not be feasible since those characteristics were not available in the dataset.

A popularity-based (measured as the number of followers for an author) metric, has all the above characteristics. In this scenario, the quality of the recommendations should be independent from the popularity of the authors (i.e., users should not be getting worse recommendations for being less popular on the platform). Concretely, users were divided into 5 quantiles (based on the number of followers of the authors in the test set). RCE and average precision was computed for each group. The final score is the average of the scoring across each group. For more details on how fairness concerns were operationalized see Section 4. The number of rows in the dataset is not equal for each cohort since more popular authors have a larger audience which typically translates into more opportunity for incoming engagement.

3 THE DATASET BEHIND THE CHALLENGE

In this section, we describe how the dataset and the challenge were formulated to address the aforementioned issues. For those familiar with Recsys Challenge 2020 [6], we will first briefly summarize the major similarities and differences from this year's version in Table 1 and then go into detail in the subsequent sections.

3.1 Dataset: features

The dataset features are described in detail in Table 2. They are divided into three separate feature groups: *user*, *tweet* and *engagement features*. There are two instantiations of *user features*, one for the author (producer) and one for the reader (consumer) of the Tweet. *Tweet features* groups all the attributes describing the original Tweet, that is possibly engaged with by the consumer. Finally, the *engagement features* contain all the details of the engagement.

2020 Challenge	**2021 Challenge**
~ 200 million records	~ 1 billion records
Scrubbing, pseudonegatives, pre-featurized text tokens	Scrubbing, pseudonegatives, pre-featured text tokens
Fully uniform sampling	Half uniform sampling and half of records sampled from a denser subgraph
Submit TSV predictions (no latency constraint)	Submit model that must run within 24h on fixed hardware
Accuracy metrics determine winner (PR-AUC and RCE)	Accuracy metrics coupled with a popularity based fairness metric determine winner

Table 1. Overview of the main differences between RecSys 2021 and RecSys 2020 Challenge

3.2 How the Dataset was built

The data for the challenge comes from a snapshot of a 4 week period on public users on Twitter. The first 3 weeks were used for the training set and the last week was used for testing and validation. We were aiming for approximately

Feature	Name	Signature	Description
User	userId	*string*	User identifier (hashed)
	follower count	*int*	Number of followers of the user
	following count	*int*	Number of accounts this user is following
	is verified	*bool*	Is the account verified?
	account creation time	*int*	timestamp of the creation time of the account
Tweet	tweetId	*string*	Tweet identifier (hashed)
	presentMedia	*list[string]*	Tab-separated list of media types; media type can be in (Photo, Video, Gif)
	presentLinks	*list[string]*	Tab-separated list of links included in the tweet (hashed)
	presentDomains	*list[string]*	Tab-separated list of domains (e.g. twitter.com) included in the tweet (hashed)
	tweetType	*string*	Tweet type, can be either Retweet, Quote, Reply, or Toplevel
	language	*string*	Identifier corresponding to inferred language of the tweet
	tweet timestamp	*int*	Unix timestamp, in seconds of the creation time of the Tweet
	tweet tokens	*list[int]*	Ordered list of Bert ids corresponding to Bert tokenization of Tweet text
	tweet hashtags	*list[string]*	Tab-separated list of hashtags present in the tweet
Engagement	reply engagement timestamp	*int*	timestamp of the Reply engagement if one exists
	retweet engagement timestamp	*int*	timestamp of the Retweet engagement if one exists
	quote engagement timestamp	*int*	timestamp of the Quote engagement if one exists
	like engagement timestamp	*int*	timestamp of the Like engagement if one exists
	engageeFollowsEngager	*bool*	Does the account of the engaged tweet author follow the account that has made the engagement?

Table 2. List of features provided for the challenge dataset

1 billion records, equally balanced between positives and (pseudo)negatives. In contrast to last year's challenge [6], we also ensured that approximately half of the positive samples came from so-called "dense users". These are users who had at least 10 incoming or outgoing engagements with other users during the time period. Before releasing the dataset, we obfuscated all the fields, especially the ones that would make the re-identification trivial (e.g. the Tweet id). We are aware that this approach is ineffective against linkage attacks (e.g. [14] and [18]) and the data can be easily reconstructed. However there are reduced privacy concerns since all the data that we started with is already public.

Pseudonegative features For the negative features we did the following: for each reader, we collected the Tweets that were produced by their followees during the sampling period. Some of those might have been (publicly) engaged with, so we removed them from this pool (they may separately have been in the positive sample). The rest of Tweets were Tweets that were not engaged with, but here is the catch: this group contains both Tweets that were seen and not seen by the reader. Negative features were sampled from this group.

Scrubbing deleted content Over time, a user might decide to delete one or more of their Tweets, or decide to delete their profile (or make it private). Again, since we want the dataset to only contain public data at all times, the dataset itself is constantly updated to track what is on the Twitter platform. Specifically this means that the dataset is shrinking as we are not adding new content over time, only deleting data that is not available anymore. Furthermore participants are required to keep their dataset up-to-date as required in the Developer Agreement and Policy and we provide a file that includes only details on the content that needs to be scrubbed.

Text Features Language is at the heart of Twitter, and for some time it was not possible to add any media to Tweets. We wanted to find a reasonable trade-off between obfuscation (i.e. including plain text would have made re-identification trivial) and making sure that the content of the Tweet would be available for NLP tasks. We released the text in form of the BERT tokens: after tokenizing the text, we release the list of the BERT IDs for each Tweet.

4 EVALUATION

In the 2020 challenge, Relative Cross Entropy (RCE) and Area under the Precision Recall Curve (PR-AUC) were adopted to evaluate each engagement. For more details on how those metrics were calculated, refer to Belli et al. [6]. In detail, the calculation of PR-AUC involved the Python `scipy` library [20]. That implementation had a downside: constant predictions got a PR-AUC of 0.5. Several teams exploited this by submitting constant predictions. To prevent the same problem, this year we adopted average precision and RCE.

Relative Cross Entropy (RCE) measures the improvement of a prediction relative to the naive prediction, measured with cross entropy (CE). In detail, we define the naive prediction as the case that does not take into account the user and Tweet features, it always predicts the average (observed) CTR of the training set. Let the average CE of the naive prediction be CE_{naive} and average CE of the prediction be CE_{pred}, then RCE is defined as $(CE_{naive} - CE_{pred}) \times 100/CE_{naive}$. It is woth mentioning that lower CE leads to better (and higher) RCE. The main advantage of adopting RCE is that it is an estimate of whether, and how much, the model is under or over performing the naive prediction.

Average Precision (AP) was implemented using the `scikit-learn` package [15]. The metric "summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight." Like PR-AUC, this metric measures accuracy and, more importantly, the constant predictions problem does not occur.

Final score In Section 2.1, we pointed out that the accuracy of the recommendations should be independent of the popularity of the author of the Tweet. We partitioned the authors in five quantiles (based on the number of followers of the authors on the test set). Thus, we have computed the average precision and RCE for each group. The two scores are then averaged across groups (in such a way that entries with good performance on popular users poor on smaller accounts are penalized) and ranked. The overall score is defined as the sum of the position for each score. For instance, one who ranked first for the RCE and third for the AP will obtain a score of 4.

5 CHALLENGE RESULTS

Table 3. Leaderboard with top-10 solutions at the end of the Challenge. RT, RE, LI and WC correspond to "Retweet", "Reply", "Like", and "with comment", respectively. AP stands for Average Precision, and RCE stands for Relative Cross-Entropy. The winning solutions for the general contest are marked with **bold**, and the winning solutions for the academic contest are marked with ***italic bold***. The last column denotes the overall score (smaller is better).

Rank	User	Method	AP RT	RCE RT	AP RE	RCE RE	AP LI	RCE LI	AP RT WC	RCE RT WC	Time	Score
1	bennys101010	**nvidia_rapidsai_final_ensemble_v2**	0.4614	29.5127	0.2649	26.6123	0.7216	23.6124	0.0692	17.6868	23 hours	2
2	_mdaniluk	**Synerise_v1**	0.4514	28.5222	0.2559	25.7468	0.7046	22.0994	0.0662	16.9245	18 hours	4
3	s_jianing	**LAYER6_AI**	0.4317	27.4239	0.249	25.3526	0.6836	19.8578	0.066	16.8696	13 hours	6
4	SamueleMeta	***test_lightgbm***	0.406	25.0928	0.2118	22.6491	0.6636	17.9193	0.052	14.0357	5 hours	8
5	perecasxiru	***final1***	0.394	24.0142	0.2077	22.1539	0.6559	16.9609	0.0459	12.6722	19 hours	10
6	sol142820574	stacking_gnn	0.3846	23.5012	0.2018	20.7757	0.6056	11.8858	0.0503	13.6293	8 hours	12
7	zhwsmile	version3-20210623-014546	0.3849	23.2816	0.1863	16.9172	0.6031	8.6238	0.0534	13.6542	9 hours	15
8	angrypark_	final_guess_2	0.3521	20.8042	0.1801	19.5385	0.6017	11.5249	0.0457	12.0744	10 hours	15
9	PanikaTM	Beta-1_1	0.3591	20.7594	0.1868	20.3206	0.5806	7.1209	0.047	12.7793	0 hours	18
10	alexkrauck	***User_Based_XGB_V9***	0.3503	18.2654	0.1786	18.6721	0.5794	8.0251	0.0406	7.5302	1 hours	21

Table 3 shows the leaderboard for the ten best solutions. The table reports the final rank, the Twitter user who submitted the winning solution, the considered metrics, the time taken to train the model(s), and finally the overall score. For what regards the metrics, an exhaustive description has already been provided in Section 4. Furthermore, the final score represents the summation of the ranks obtained in the other metrics and, therefore, smaller is better. With respect to approaches, it could be observed that almost all the winning solutions deeply analyzed and exploited the methods that won the 2020 competition. As a recall, most of the solutions were a blend of machine learning techniques, and all the winning solutions devoted much effort to feature engineering, considering it a key aspect of the competition. The considered recommendation families ranged from graph-based to knowledge-aware [2], time-aware [16], spatial [3, 19], device-aware [4] embedding-based [10], community-based [9].

6 THE WINNER SOLUTIONS

One of the most important differences with respect to RecSys 2020 Challenge is how to define which are the winning solutions. To fairly consider the resources available in Academia and Industry, we have identified three winning solutions from the general leaderboard and the three best solutions whose teams are composed only of academic people. In the following, we provide a brief overview of the **six winning methods**.

1st Place - GPU Accelerated Boosted Trees and Deep Neural Networks for Better Recommender Systems The authors proposed an end-to-end GPU-accelerated ensemble of stacked models, using in total 5 XGBoost models and 3 neural networks. The authors performed Feature Engineering to extract meaningful respresentative features using Target Encoding, which calculates the conditional probabilities of the interaction given sets of categorical features. The authors used gradient boosted trees (XGBoost) and neural networks for predictions.

2nd Place - Synerise at RecSys 2021: Twitter user engagement prediction with a fast neural model The authors realized a simple feed forward neural network that predicts the probability of different engagements types for user and target tweet. However, they first obtain tweet text representation by fine-tuning a DistilBERT model [8] on tweets, and then use EMDE [3] to represent text as a sketch (a compressed, fixed-size representation of the tweet meaning). Finally, they train the model in two stages: i) using training set, and then ii) fine-tuning the model on validation set.

3rd Place - User Engagement Modeling with Deep Learning and Language Models The authors proposed a hybrid Java and Python pipeline to extract features from Tweets content which are valuable towards user engagement. The engineered features are used to train 4 XGBoost models, one for each engagement type. The authors trained a neural classifier with multi-layer perceptions (MLPs) to predict on users engagement probabilities. Extracted features are used as input to the neural classifier and are mapped to a 4-dimensional prediction logits through a feedforward neural networks and a fully-connected layer.

1st Academic Place - Lightweight and Scalable Model for Tweet Engagements Predictions in a Resource-constrained Environment The authors proposed an optimized LightGBM model, which leverages a wide variety of meaningful 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. The authors adopted two types of models for engagement prediction: Neural Network (NN) and Gradient Boosting for Decision Tree (GBDT) models.

2nd Academic Place - Addressing the cold-start problem with a two-branch architecture for fair tweet recommendation They presented a two-branch architecture that separates Twitter authors according to their total number of interactions in the dataset. To clarify, the authors who appear a few number of times (cold-start users) are predicted using similar users. The same holds for users with many interactions (warm users). For each of the two branches, we have designed a concatenation of LightGBM models that, trained independently, use the predictions of the easiest targets (Like and Retweet), to predict the more complex targets (Reply and Quote). In their experience, the users' popularity, as well as the first and last words of the tweet text, turned out to be the best features.

3rd Academic Place - Team JKU-AIWarriors in the ACM Recommender Systems Challenge 2021: Lightweight XGBoost Recommendation Approach Leveraging User Features The authors proposed a model that relies on features that can be computed from user engagement counts. These counts are used to create compact user-specific features, which enables the model to make predictions quickly. They exploit a simple XGB classifier, trained on a subset of the training data. To regularize during training, they added Gauss-distributed noise and randomly masked users to avoid overfitting. Their approach is so efficient that took less than a tenth of the average runtime of the nine better performing approaches.

REFERENCES

[1] Fabian Abel, Yashar Deldjoo, Mehdi Elahi, and Daniel Kohlsdorf. 2017. RecSys Challenge 2017: Offline and Online Evaluation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) *(RecSys '17)*. Association for Computing Machinery, New York, NY, USA, 372–373. https://doi.org/10.1145/3109859.3109954

[2] Vito Walter Anelli, Pierpaolo Basile, Derek G. Bridge, Tommaso Di Noia, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Markus Zanker. 2018. Knowledge-aware and conversational recommender systems. In *RecSys*. ACM, 521–522.

[3] Vito Walter Anelli, Andrea Calì, Tommaso Di Noia, Matteo Palmonari, and Azzurra Ragone. 2016. Exposing Open Street Map in the Linked Data Cloud. In *Trends in Applied Knowledge-Based Systems and Data Science - 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2016, Morioka, Japan, August 2-4, 2016, Proceedings (Lecture Notes in Computer Science, Vol. 9799)*, Hamido Fujita, Moonis Ali, Ali Selamat, Jun Sasaki, and Masaki Kurematsu (Eds.). Springer, 344–355. https://doi.org/10.1007/978-3-319-42007-3_29

[4] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, and Antonio Ferrara. 2019. Towards Effective Device-Aware Federated Learning. In *AI*IA (Lecture Notes in Computer Science, Vol. 11946)*. Springer, 477–491.

[5] Vito Walter Anelli, Amra Delić, Gabriele Sottocornola, Jessie Smith, Nazareno Andrade, Luca Belli, Michael Bronstein, Akshay Gupta, Sofia Ira Ktena, Alexandre Lung-Yut-Fong, et al. 2020. RecSys 2020 Challenge Workshop: Engagement Prediction on Twitter's Home Timeline. In *Fourteenth ACM Conference on Recommender Systems*. 623–627.

[6] Luca Belli, Sofia Ira Ktena, Alykhan Tejani, Alexandre Lung-Yut-Fon, Frank Portman, Xiao Zhu, Yuanpu Xie, Akshay Gupta, Michael Bronstein, Amra Delić, Gabriele Sottocornola, Walter Anelli, Nazareno Andrade, Jessie Smith, and Wenzhe Shi. 2020. Privacy-Aware Recommender Systems Challenge on Twitter's Home Timeline. arXiv:2004.13715 [cs.SI]

[7] Ching-Wei Chen, Paul Lamere, Markus Schedl, and Hamed Zamani. 2018. Recsys Challenge 2018: Automatic Music Playlist Continuation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. Association for Computing Machinery, New York, NY, USA, 527–528. https://doi.org/10.1145/3240323.3240342

[8] Yashar Deldjoo, Vito Walter Anelli, Hamed Zamani, Alejandro Bellogin, and Tommaso Di Noia. 2020. A flexible framework for evaluating user and item fairness in recommender systems. *User Modeling and User-Adapted Interaction* (2020), 1–47.

[9] Amra Delic, Judith Masthoff, Julia Neidhardt, and Hannes Werthner. 2018. How to Use Social Relationships in Group Recommenders: Empirical Evidence. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization, UMAP 2018, Singapore, July 08-11, 2018*, Tanja Mitrovic, Jie Zhang, Li Chen, and David Chin (Eds.). ACM, 121–129. https://doi.org/10.1145/3209219.3209226

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[11] Dylan Hadfield-Menell and Gillian K. Hadfield. 2019. Incomplete Contracting and AI Alignment. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (Honolulu, HI, USA) *(AIES '19)*. Association for Computing Machinery, New York, NY, USA, 417–422. https://doi.org/10.1145/3306618.3314250

[12] Peter Knees, Yashar Deldjoo, Farshad Bakhshandegan Moghaddam, Jens Adamczak, Gerard-Paul Leyson, and Philipp Monreal. 2019. RecSys Challenge 2019: Session-Based Hotel Recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 570–571. https://doi.org/10.1145/3298689.3346974

[13] Smitha Milli, Luca Belli, and Moritz Hardt. 2021. From Optimizing Engagement to Measuring Value. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (Virtual Event, Canada) *(FAccT '21)*. Association for Computing Machinery, New York, NY, USA, 714–722. https://doi.org/10.1145/3442188.3445933

[14] Arvind Narayanan and Vitaly Shmatikov. 2006. How To Break Anonymity of the Netflix Prize Dataset. arXiv:cs/0610105 [cs.CR]

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[16] Pablo Sánchez and Alejandro Bellogín. 2018. Time-Aware Novelty Metrics for Recommender Systems. In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 10772)*, Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury (Eds.). Springer, 357–370. https://doi.org/10.1007/978-3-319-76941-7_27

[17] Andrew D. Selbst, Danah Boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. Fairness and Abstraction in Sociotechnical Systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 59–68. https://doi.org/10.1145/3287560.3287598

[18] Latanya Sweeney. 1997. Guaranteeing anonymity when sharing medical data, the Datafly System. In *Proceedings: a conference of the American Medical Informatics Association. AMIA Fall Symposium*. Hanley & Belfus, Inc., Nashville, TN, USA, 51—55. https://europepmc.org/articles/PMC2233452

[19] Sergio Torrijos, Alejandro Bellogín, and Pablo Sánchez. 2020. Discovering Related Users in Location-based Social Networks. In *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2020, Genoa, Italy, July 12-18, 2020*, Tsvi Kuflik, Ilaria Torre, Robin Burke, and Cristina Gena (Eds.). ACM, 353–357. https://doi.org/10.1145/3340631.3394882

[20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert

Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2