Detecting and Limiting Negative User Experiences in Social Media Platforms

Luis Garcia Pueyo
lgp@meta.com
Meta Platforms, Inc.

Vinodh Kumar Sunkara
vinodhsunkara@meta.com
Meta Platforms, Inc.

Prathyusha Senthil Kumar
prathyushas@meta.com
Meta Platforms, Inc.

Mohit Diwan
mohitd@meta.com
Meta Platforms, Inc.

Qian Ge
qge2@meta.com
Meta Platforms, Inc.

Behrang Javaherian
bjavaherian@meta.com
Meta Platforms, Inc.

Vasilis Verroios
verroios@meta.com
Meta Platforms, Inc.

ABSTRACT

Item ranking is important to a social media platform's success. The order in which posts, videos, messages, comments, ads, used products, notifications are presented to a user greatly affects the time spent on the platform, how often they visit it, how much they interact with each other, and the quantity and quality of the content they post. To this end, item ranking algorithms use models that predict the likelihood of different events, e.g., the user liking, sharing, commenting on a video, clicking/converting on an ad, or opening the platform's app from a notification. Unfortunately, by solely relying on such event-prediction models, social media platforms tend to over optimize for short-term objectives and ignore the long-term effects. In this paper, we propose an approach that aims at improving item ranking long-term impact. The approach primarily relies on an ML model that predicts negative user experiences. The model utilizes all available UI events: the details of an action can reveal how positive or negative the user experience has been; for example, a user writing a lengthy report asking for a given video to be taken down, likely had a very negative experience. Furthermore, the model takes into account detected integrity (e.g., hostile speech or graphic violence) and quality (e.g., click or engagement bait) issues with the content. Note that those issues can be perceived very differently from different users. Therefore, developing a personalized model, where a prediction refers to a specific user for a specific piece of content at a specific point in time, is a fundamental design choice in our approach. Besides the personalized ML model, our approach consists of two more pieces: (a) the way the personalized model is integrated with an item ranking algorithm and (b) the metrics, methodology, and success criteria for the long term impact of detecting and limiting negative user experiences. Our evaluation process uses extensive A/B testing on the Facebook platform: we compare the impact of our approach in treatment groups against production control groups. The AB test results indicate a 5% to 50% reduction in hides, reports, and submitted feedback. Furthermore, we compare against a baseline that does not include some of the crucial elements of our approach: the comparison shows our approach has a 100x to 30x lower False Positive Ratio than a baseline. Lastly, we present the results from a large scale survey, where we observe a statistically significant improvement of 3 to 6 percent in users' sentiment regarding content suffering from nudity, clickbait, false / misleading, witnessing-hate, and violence issues.

WWW '23, April 30–May 04, 2023, Austin, TX, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9416-1/23/04.
https://doi.org/10.1145/3543507.3583883

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Machine learning**; • **Information systems** → **Social networks**.

KEYWORDS

machine learning, personalization, social networks, integrity

ACM Reference Format:
Luis Garcia Pueyo, Vinodh Kumar Sunkara, Prathyusha Senthil Kumar, Mohit Diwan, Qian Ge, Behrang Javaherian, and Vasilis Verroios. 2023. Detecting and Limiting Negative User Experiences in Social Media Platforms. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3543507.3583883

1 INTRODUCTION

Item ranking is one of the most fundamental problems in social media platforms. The items can be posts or videos uploaded by platform content creators or a user's friends and presented in ranked order in the user's feed, comments appearing in the comments section for a given post or video, ads posted by companies or platform users for new or used products being sold, or notification candidate messages, which need to be sorted, so that the top one or two messages form the actual notifications reaching a user. Item ranking in those (and many other) scenarios greatly affects how much time users spent on the platform, how often they interact with other users and friends, how often they use the platform, how often they buy or sell products in a used items marketplace, the type, quality, and amount of content professional creators produce, and so on. With those goals in mind, ranking algorithms focus on the prediction of events like sharing, commenting-on, or liking a post, clicking on an ad or notification, or using an augmented reality filter they saw in a video.

As an example consider a very simple ranking algorithm, relying on only three predictions: the user liking, sharing, or hiding a given item. In this example, items are ranked, in descending order, based on a linear formula: $w_1 * p(Like) + w_2 * p(Share) - w_3 * p(Hide)$. Most platforms would decide weights w_1 to w_3 based on an offline analysis that would indicate how a share should be valued against a hide or like. For instance, if $w_1 = 1$, $w_2 = 2$, and $w_3 = 0.5$, the ecosystem value of a like is equal to half the value of a share, and twice the value of a hide.

Unfortunately, item ranking mainly driven by such event predictions, often becomes too optimized for short term goals, while ignoring the long-term impact: lower quality/integrity content (e.g., click bait, graphic violence, gross, sexually suggestive, misinformation, bullying, hate) gets promoted and users' perception about the platform and the platform's content changes, until they eventually end up using the platform much less often, delete their account, or, even worse, develop negative emotions when using the platform.

In this paper, we focus on the *long-term* impact of item ranking in social media platforms, and specifically on the problem of multi-objective item ranking: one set of objectives comes from the *short-term* engagement goals and the second set focuses on limiting *negative user experiences.* Those negative experiences take a heavy toll on users' emotional health and affect engagement and society's perception about a platform, in the *long run.*

The main challenge for multi-objective ranking is how different users' perception can vary about content that suffers from quality or integrity issues: borderline violence, nudity, profanity can go unnoticed by some users or shock others. And there is also a large portion of users that will find such content very engaging. The same applies to a vast set of content classes related to quality and integrity issues like misinformation (consider a sports article focusing on a critical referee decision or a post with health related opinions), spam or bait content, or content reproduced from the web or external platforms. The naive approach is to try and limit such content by uniformly demoting it for all users. However, by being too strict on what should be considered a quality/integrity issue, the uniform demotion of such content ends up having an extensive negative impact on the engagement metrics for a platform; by being too loose, on the other hand, the negative long term impact still remains. The problem with this naive approach is the very different perspective of users on quality and integrity and the fact that many negative experiences are caused by content that cannot be directly classified as a quality/integrity issue; for instance, consider a pets training video, where someone could find the training method cruel.

Another naive approach is to introduce simple event predictions to capture negative user engagement, like the $p(Hide)$ factor, in the aforementioned example. Unfortunately such simple predictions alone, cannot fully capture negative experiences: users may rely on such UIs, like a Hide button, for many different reasons. To deeply understand and detect negative experiences, a platform needs to rely on all available signals from a user's activity: we provide more evidence supporting this statement, when discussing our approach and in the experiments section.

The larger the social media platform, in terms of user base and content inventory, the more dramatic the long-term impact of short-term optimized ranking. Moreover, long-term might mean quarters or years until a drop in user retention is observed, other significant engagement metrics are affected, or the platform realizes a shift in users' and society's perception on the platform's content and social value. The necessary long observation period and platform's size are the main factors explaining why this problem has not been effectively solved so far: the approach we propose relies on long-term analysis and findings in one of the largest social media platforms, with an extensive user adoption globally.

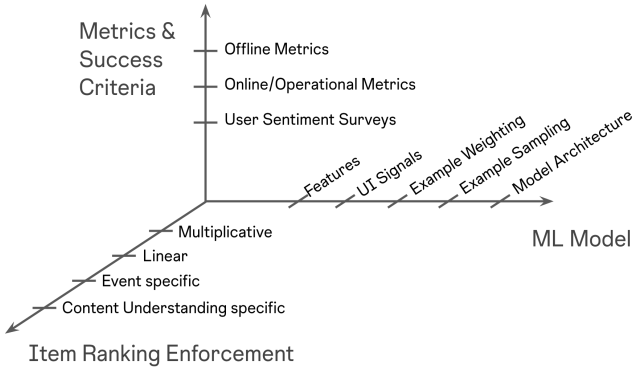

Figure 1: Key elements of our approach

Our approach consists of three elements: (1) a personalized ML model predicting negative user experiences, (2) item ranking integration, which captures how the personalized model can be combined with different item ranking algorithms and (3) a framework covering metrics, methodology, and success criteria for the long term impact of detecting and limiting negative user experiences. Through our experiments we observe a reduction of 5 to 50 percent in hides, reports, and submitted feedback, when introducing the ML model and item ranking integration in different sections of the Facebook app. Moreover, we see 100x to 30x improvements in the False Positive Ratio and Like-Through-Rate, when comparing to a baseline, and a statistically significant improvement of 3 to 6 percent in users' sentiment related to content suffering from integrity issues, based on a large scale survey.

The rest of the paper is organized as follows:

- ML model and ranking algorithm: technical details on the ML model and the model's integration with ranking algorithms.
- Metrics and Success Criteria: generic framework to assess the long-impact of item ranking solutions with both engagement and negative user experience objectives.
- Experiments: extensive A/B test and survey results quantifying the impact of our approach in multiple sections of the Facebook app, where item ranking is applied.
- Related Work: discussion on previous research in the areas of user understanding, personalization, content understanding, and integrity-harm detection in social media platforms.

2 APPROACH

Figure 1 summarizes the three elements of our approach: we discuss the ML model and the item ranking algorithm integration in this section, after we first provide some context on the objective and subjective integrity issues, social media platforms face.

On the ML model, we focus on:

(1) the training data: how we leverage UI signals and how we sample and weigh the resulting training examples, to infer strong negative and positive user experiences,

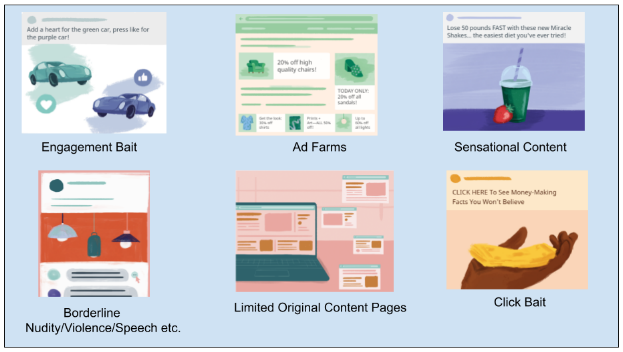
Add a heart for the green car, press like for the purple car!
Engagement Bait
20% off high quality chairs!
TODAY ONLY: 20% off all sandals!
Get the look: 30% off shirts
Prints + Art—ALL 50% off!
Up to 60% off all lights
Ad Farms
Lose 50 pounds FAST with these new Miracle Shakes... the easiest diet you've ever tried!
Sensational Content
Borderline Nudity/Violence/Speech etc.
Limited Original Content Pages
CLICK HERE To See Money-Making Facts You Won't Believe
Click Bait

Figure 2: Low Integrity Content

(2) the features: we discuss the types of features used to predict negative and positive user experiences, and
(3) the model type and architecture: we discuss the model type and architecture alternatives and the design choices we made in the models we developed.

On the item ranking integration, we discuss the different ways the negative user experience prediction model can be leveraged by ranking algorithms and the corresponding trade-offs.

2.1 Context

2.1.1 Objective Integrity Issues. Social media platforms typically use objective criteria on the type of content allowed, e.g., Facebook Community Standards [3]. Examples of such content include Violence and Criminal behavior, Child and Adult Safety Issues, Weapons and firearms, etc. Beyond those policy-violating content types, there are also types of low quality/integrity content that are not explicitly blocked by platforms, but are still problematic and often cause negative user experiences [1]. Figure 2 gives examples of such problematic content types: posts with links to disruptive ads, posts seeking engagement, posts with sensational claims like miracle cures for diseases, borderline nudity, etc. Those content types are considered problematic regardless of the user viewing them, and social media platforms usually demote them equally, for all users, when applying an item ranking algorithm.

2.1.2 Subjective Integrity and Quality Issues. When going over the posts, videos, and articles in social media platforms, users may have a negative experience with any type of content. Borderline content that nearly matches but does not violate policy standards, is often the cause of a negative experience. But most of the negative user experiences are usually caused by content that is not even a borderline policy violation. For example, a steak-cooking or rodeo video might be considered repulsive by users based on their culture or personal preferences and beliefs (e.g., on animal cruelty).

2.2 ML Model

The ML model predicts if a given user, viewing a given item (e.g., an ad, video, post, etc.), at a given point in time, will have a negative or positive experience.

2.2.1 Training Data. The model leverages all available UI signals to infer how negative or positive a user experience has been. We are using past user sessions, where in each session, a user is presented with a list of items (e.g., posts in a feed), scrolls over the items, and takes action on a subset of them. Positive engagement UI actions like sharing, liking a post, opening the comments section, or using a music track or AR filter from a video, are treated as negative examples in a classification problem (we discuss more alternatives in Section 2.2.3). The positive examples are negative engagement UI actions like hiding or reporting a post. The specific UI actions depend on the interface controls of the platform: in our case, we have developed models for different sections of the Facebook app and utilized the UI controls of each section for the respective model.

- Example Sampling: training examples are sampled based on
  - action frequency: rare actions like reports are upsampled,
  - user's action statistics: examples from power users or heavy feedback providers are downsampled, in order to avoid model overfitting on specific users' behaviors,
  - detected content integrity or quality issues: examples with a detected integrity or quality issue are upsampled; such content is more likely to trigger a conscious negative-experience reaction for a user that hides or reports that piece of content.
- Example Weights: a weight is assigned to each example to capture how negative or positive each experience has been; the more negative or positive the experience, the higher the weight. We rely on a number of proxies for the level of discomfort or excitement a post, video, ad might have caused to a user:
  - the statistical correlation of each UI action with survey results (more on the survey design in Section 3.3),
  - the time required to take action by a user (e.g., writing and submitting a report takes much more time than hiding a post in most UIs), and
  - the frequency of the UI action.

2.2.2 Features. The model features can be organized into the following categories:

Item level: information regarding the item being ranked (for a target user). This information can be based on the item's metadata (e.g., video length, post age) or the output of other models on the item's image, video, audio, or text data (e.g., detected language). Besides common information about the item, there are three types of item-level features that are particularly important:

(1) Item Statistics: counts and rates of any UI action applied to the item, by all users, e.g., number of likes, shares, comments, creator-profile open, like-through-rate, share-through-rate, hide-rate, etc. Those statistics are collected over different time windows, e.g., last hour, day, week, month, and so on, and they might also be collected over different user groups, e.g., country, language, age-group, stats.
(2) Item Embeddings: one critical design choice in our model is the use of content embeddings, generated by other, generic, models, instead of using raw video, image, audio, or text data. In our case, the generic embedding models are trained over both engagement and integrity objectives and are used by multiple item-ranking models across the different sections of the app, e.g., models predicting the likelihood of a user liking, sharing, or disliking a video.
    - Trade-off between using content embeddings or raw data: the obvious trade-off is related to the amount of training

data. Training over raw data requires a lot more training examples to avoid overfitting, however, it can improve a specific model's accuracy, if the right amount of examples is available. In large social media platforms, a sufficiently large amount of examples is not the main challenge. The main challenge is computational cost. For each user opening the app, in every section of the app where item ranking is applied, the app's backend runs model inference for every candidate item (in the order of thousands or tens of thousands, depending on the ranking stage a model is applied at). That online cost is the main reason (offline/training cost can also be substantial) for keeping models as lightweight as possible: content embeddings significantly help reduce both the offline and, primarily, the online computational cost.

(3) Item content classification: besides content embeddings, the negative experience model uses the predictions from content classification models on specific integrity harms, e.g., nudity, graphic violence, misinformation, bait, profanity, hostile speech, bullying, blurry image, non-original content etc. Those content classes are highly correlated with negative user experiences.

User level: information regarding the target user, e.g., user age, country, languages, date of joining the platform, number of friends, etc. Statistics and embeddings are again critical for the negative experiences model:

(1) User statistics: counts and rates on actions on the platform, from this user. The statistics are collected over different time windows, e.g., last hour, day, week, month, and they might be grouped over different sections on the app, e.g., statistics on the Home tab, or different content classes, e.g., statistics on pets, food, dance, cars, etc.
(2) User embeddings: capture user's preferences via latent dimensions; in our case, we have used both: (a) user embeddings that were co-trained with content embeddings and (b) transformer models (transformer [15] and linformer [16]) that train over user sessions, where each session is represented by a sequence of (item, user-action) pairs, i.e., the action taken by a given user on a given item.

User-Item level: information regarding the given item for the target user, e.g., if the user has seen this item before and any likes or shares during previous interactions of the user with the item. The following statistics are some of the most important user-item specific statistics; all statistics are collected over time windows, e.g., last hour, day, week, month:

(1) Item statistics from similar users: counts and rates on UI actions, for the item in question, from the target user's country, age group, language, or user friends.
(2) User-Creator/Group level statistics: counts and rates on UI actions, from the target user on the item's creator (e.g., friend, publisher, or advertiser) or group (in case creators can collectively publish content in a group).
(3) User-Topic level statistics: counts and rates on UI actions, from the target user on the item's topic; topics can be content classes detected on the item or annotations provided by other users.

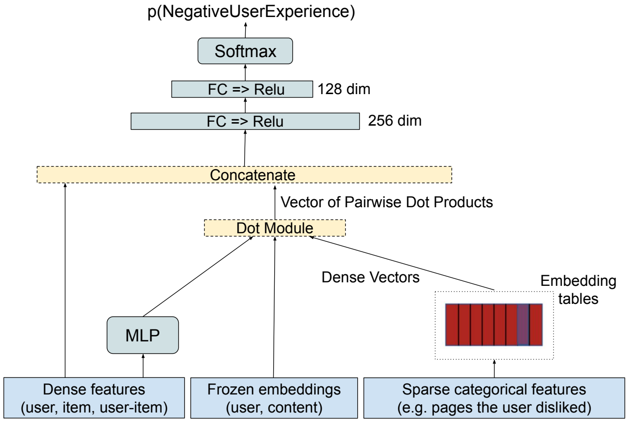
p(NegativeUserExperience)
Softmax
FC => Relu
128 dim
FC => Relu
256 dim
Concatenate
Vector of Pairwise Dot Products
Dot Module
Dense Vectors
Embedding tables
MLP
Dense features (user, item, user-item)
Frozen embeddings (user, content)
Sparse categorical features (e.g. pages the user disliked)

Figure 3: Connected feed model architecture

2.2.3 Model Type and Architecture. From a model architecture point of view, the features fall under three categories:

(1) Imported / Frozen embeddings: embeddings trained by external models (see Section 2.2.2) on the target user, the target item, the item's creator, the item's group, the item's music track, or the item's AR filter. Those embeddings are frozen during training, i.e., no gradient updates over the vectors' values.
(2) Categorical / Non-frozen embeddings: categorical features are handled as non-frozen embeddings, i.e., lookup from category id to a vector trained during the overall model's gradient descent. Examples of categorical features handled as non-frozen embeddings are the user's language and country.
(3) Dense features: statistics and other non-categorical, non-sparse features.

Figure 3 depicts the details for the negative experience model developed for the Connected feed section of the Facebook app (where users can see content from friends and accounts they follow). The implementation is based on an internal development framework for deep learning models [9] built on top of pytorch and caffe2. The framework's modularization components provide flexibility and allow us to reuse pieces across the models of the different sections of the Facebook platform. In Figure 3, you see a Sparse module taking in categorical features and converting them into dense vectors through embedding tables. Dense features go through an MLP module, which outputs vectors of the same number of dimensions with the frozen and non-frozen embeddings. All vectors are combined via dot products: a vector of dot products is concatenated with dense features and passed into the final layers consisting of fully connected linear / ReLu modules and a Softmax negative experience prediction. We have found this combination of embeddings, dot product elements, and linear layers, to capture well non-linear correlations between features, while keeping the inference computational cost low.

Due to the unique characteristics and UI of different sections of the Facebook platform, the risk of overfitting to the data of specific sections, over time, and several computational cost factors, we have developed separate models for different sections. All section models rely on the same high level architecture with the one in Figure 3. We present results from AB tests in the different sections of the app in Section 4.2.

Model Type Alternatives: modeling user negative experiences as a classification problem with weights and sampling over the different examples has been very effective in our case, primarily due to the simplicity of monitoring the different sections' models (e.g., accuracy, feature importance) over time. For completeness, we would like to mention a couple of other approaches to model the negative user experience problem:

- Regression Formulation: example weights and sampling can be combined into a single value, a regression model can be trained to predict; the negative user experience model then predicts a scalar value: the higher that value, the more negative the predicted user experience.
- Partial-order Formulation: in this case, examples are not assigned a specific value, but they are ordered based on how negative or positive the experience of the user has been. The partial order can be constructed at a user or session level, or simply capture the different types of actions, e.g., report > hide > like > share > mimicry (e.g., create a video with the same music track or AR filter), from most negative to most positive.

2.2.4 Item ranking enforcement. There are four main ways negative user experience predictions can be integrated into an item ranking algorithm:

1. Multiplicative: the Negative User Experience prediction ($p(NUE)$) is applied as a multiplicative factor to each item's value. In the example in the introduction, item ranking could be extended to

$$[1 - p(NUE)] * [p(Like) + 2 * p(Share) - 0.5 * p(Hide)]$$

   One main assumption for the correctness of the multiplicative factor is the pre-existing item values being positive: in that case, $[1 - p(NUE)]$ acts as a demotion factor that downranks items with a high $p(NUE)$ value. In practice, demotion curves can be non-linear and use a threshold for the $p(NUE)$ prediction; below that threshold no demotion is applied.
2. Linear: the $p(NUE)$ prediction is applied as a linear factor in the item value formula. In the introduction's example, item ranking could change to $p(Like) + 2 * p(Share) - 0.5 * p(Hide) - 1.5 * p(NUE)$; i.e., an expected negative user experience corresponds to 1.5 expected likes, three (1.5/0.5) hides, and 0.75 shares.
   - Trade-off between multiplicative and linear enforcements: linear enforcements add less complexity (e.g., the NUE prediction is treated like any other prediction; no non-negative item value assumptions required) and are more interpretable. Multiplicative enforcements' main advantage is that their impact is not affected by changes in the rest of the item ranking components, e.g., if the weights of p(Like), p(Share), and p(Hide) were increased by a factor of 10, in the introduction's example, multiplying item values by $[1 - p(NUE)]$ would have the same re-ordering impact.
3. Event specific: in this case, the $p(NUE)$ prediction only affects a subset of the events in an item ranking formula. For example, the $[1 - p(NUE)]$ factor could be multiplied with a $p(Comment)$ prediction, to indicate that an expected comment should be weighted by the likelihood of a positive experience leading to that comment.
4. Content Understanding specific: more aggressive demotion curves and $p(NUE)$ thresholds for specific content classes, especially, for the classes indicating integrity harms and quality issues.

In our case, we use all four types of enforcements in different sections of the Facebook app; in some sections we use a combination of two or more enforcements.

3 METRICS AND SUCCESS CRITERIA

The subjective nature of the negative user experience problem and the fact that social media platforms only have access to implicit user signals (e.g., liking, sharing, or hiding) rather than explicit experience ratings, makes the evaluation of any approach challenging. The assessment framework we propose consists of three components

3.1 Offline predictive accuracy metrics

There are a few important details when computing offline accuracy metrics for a negative user experience classification model. Since both the set of positive and negative examples derive from multiple UI signals, which have different frequencies, we propose the following sets of metrics:

1. Unweighted Precision-Recall (PR) and ROC AUC (Area Under the Curve): this is the simplest version of the PR and ROC AUC, with each example having the same weight. Nevertheless, the testing set can still be generated by the same sampling strategy as the training set.
2. UI-specific PR and ROC AUC: the accuracy for each UI signal used in the examples, is assessed in isolation. For example, consider a model using dislikes and hides for negative examples and likes and shares for positive examples. There are eight PR and ROC curves that can be generated: (1) negatives: dislikes - positives: likes and shares, (2) negatives: hides - positives: likes and shares, (3) negatives: hides and dislikes - positives: likes, (4) negatives: hides and dislikes - positives: shares, (5) negatives: dislikes - positives: likes, (6) negatives: dislikes - positives: shares, (7) negatives: hides - positives: likes, and (8) negatives: hides - positives: shares.
3. User weighted PR and ROC AUC: curves are generated per user and a single PR and ROC AUC is computed by weight averaging the per user AUCs. Both the unweighted and the UI specific, per user, PR and ROC curves can be used depending on the goals of the assessment.

Metrics' Purpose: the offline metrics are not meant to assess how well negative user experiences are captured by a model. Their sole purpose is to compare two negative experience classification models: in our case, we use offline metrics when introducing improvements on a model (e.g., new features or architecture), when we do modeling changes (e.g., different UI signal weights), or when UI changes affect the label or feature distributions, in different sections of the platform; depending on the change, we might use a subset of the offline metrics above.

3.2 AB-testing methodology

No enforcement AB tests: the treatment group has the negative experience model predictions and item ranking values computed for the group's users. However, the actual ranking of items remains the same as in the control group: by computing the hypothetical enforcement, we are able to annotate views of items that would have been impacted, i.e., demoted, for each user in the treatment group. Separate annotations are used for different prediction thresholds: the annotation data helps in tuning of the enforcement demotion curve and threshold. The no-enforcement AB also helps in quantifying any increase in computational cost (no negative experience model inference in control) and, more importantly, in collecting the following metrics:

(1) False Positive Rate (FPR) on impacted views: FPR is the ratio of False Positives (FP) to the sum of False Positives (FP) and True Negatives (TN), $FPR = FP/(FP + TN)$. False positives are all the likes, shares, and clearly positive experience actions, users would take on annotated views/items that would have been demoted by the enforcement. True Negatives refer to the same set of actions on views that were **not** demoted. The rationale of the FPR metric is to quantify the collateral damage of an approach and how often items get wrongfully demoted.
(2) Like and Dislike Through Rate (LTR and DLTR) on impacted views: note that the denominator here is all impacted views, as opposed to the union of FPs and TNs, used in the FPR denominator. The likes' set can include any positive experience actions besides likes (e.g., shares), while the dislikes' set can include negative experience actions, e.g., hides and reports. LTR and DLTR are proxies for the effectiveness of personalized demotions. An aggressive demotion impacting a large set of views, which has a very low LTR and very high DLTR, proves that the demotion targets the right type of content for the right users. Moreover, the LTR and DLTR on the impacted views must be compared with the overall LTR and DLTR in the section of the platform, the demotion applies.
(3) Impacted Views Set Size: the larger the set, the bigger the impact (positive or negative) of the approach. This metric allows us to quantify and adjust the reach of a model/enforcement.
(4) Overlap with other models and item ranking enforcements: jaccard similarity of the impacted-view sets of two different models/enforcements. For instance, we might want to quantify to what extent a p(Hide) and a negative experiences model, demote the exact same views, at high prediction thresholds. The overlap metrics can be extended to the sets of impacted likes, dislikes, shares, reports, hides, etc.

Metrics' Purpose: The metrics above identify a number of important model and enforcement properties: low collateral damage, effective personalization, high impact, and limited overlap with existing models and enforcements.

Actual enforcement AB tests: the only difference with the no-enforcement AB tests, is that the treatment group in this case, actually changes the ranking by applying the demotion, instead of just annotating potential demotions. The metrics from such AB tests focus on the increase/decrease, compared to control, in reports, feedback, hides, likes, shares, time spent on the different sections of the app, etc. Those metrics are monitored overall and for specific content classes, e.g., integrity harms like graphic violence, hostile speech, or bait, and user slices, e.g., age groups, rare vs power users, etc. Moreover, such AB tests are also setup as long-term holdouts, to assess the long-term impact of a negative user experience model through user retention metrics, e.g., daily active users or users spending more than 10 minutes per day in a section of the app.

3.3 User sentiment surveys

Rationale: surveys provide a complimentary, to AB testing, method for assessing how well an approach captures negative user experiences: through multiple runs of AB tests and surveys, the connection between AB test metrics statistical significant movement and survey results can be inferred. This connection allows for a better understanding of the overall sentiment impact and how sentiment affects operational metrics, in the long run.

Design: a survey consists of a top-level question asking if users had a bad experience recently, and branch questions asking about specific integrity issues. Aggregating survey responses across branch questions helps in measuring the reach and intensity of each integrity issue. Branch questions are composed of two parts:

- Reach question: Have you ever seen <content-type> on Facebook that you didn't want to see? – Binary "Yes/No" question. The <content-type> part is replaced by one of the integrity harms prevalent in the Facebook app, like Ad Farms, Click Bait etc. Answering "Yes" leads to the Intensity question.
- Intensity question: How bad does this experience make you feel? – A five-grade-scale question: Very bad, Pretty Bad, Moderately bad, Slightly bad, Not at all bad.

4 EXPERIMENTS

We have organized the experimental results into three sections:

Negative experience model vs a simple event prediction model: we start with an interesting baseline evaluation, where we compare the impact from a negative user experience model to the impact of a simple event prediction model. To best illustrate the impact difference, we use a no-enforcement AB test, and the impacted views, FPR, LTR, and DLTR metrics. The comparison takes place in the "Connected" feed of the Facebook app, i.e., the feed section where users see content from friends and users or groups they follow.

Impact of negative experience models across the Facebook app: we discuss the impact a negative user experience model has, when introduced in different sections of the app, namely, Connected feed, Recommendations, Videos, and Reels; Recommendations refers to the recommended content a user can experience in their feed, from creators or groups they do not follow, Videos is the long-form video feed section, and Reels refers to a short-form (e.g., less than a minute) video section in the app. We use a separate actual enforcement AB test for each section.

User sentiment survey results: we ran a user sentiment survey focusing on Ad Farms, Quality, Nudity, Click Bait, False / Misleading, Profanity, Bullied Witness, Hate Witness, and Violence (see Section 3.3). The survey is performed on two groups of users: in the treatment users have a negative experience model deployed in

Compare operational metrics at different Impacted Views percentages for Event vs Negative-Experience Model										
Impacted Views (based on prediction thresholds)	0.50%	1%	1.50%	2%	2.50%	3%	3.50%	4%	4.50%	5%
DLTR Ratio: Experiment DLTR over section's average DLTR										
Event Model	66.32	48.42	38.95	33.68	27.37	23.16	20.00	17.89	16.84	14.74
Negative Experience Model	30.53	21.05	15.79	13.68	11.58	10.53	9.47	8.42	7.89	7.37
FPR: Ratio of FP over (FP+TN)										
Event Model	0.0099	0.0133	0.0167	0.0196	0.0235	0.0286	0.0322	0.0370	0.0396	0.0437
Negative Experience Model	0.0001	0.0002	0.0004	0.0005	0.0006	0.0008	0.0009	0.0011	0.0013	0.0014
LTR Ratio: Experiment LTR over section's average LTR										
Event Model	1.144	1.060	1.022	0.977	0.963	0.939	0.932	0.921	0.907	0.897
Negative Experience Model	0.0195	0.0212	0.0226	0.0229	0.0236	0.0243	0.0257	0.0261	0.0268	0.0275

Table 1: Metrics from Section 3.2 comparing *Event* vs *Negative Experience* Model

their Connected feed, while in control users do not have a negative experience model in any section of the Facebook app.

4.1 Negative experience model vs a simple event prediction model

We compare a simple negative user experience model with a simple event prediction model. Our goal is to illustrate how much of a difference even a small subset of the modeling design choices, discussed in Section 2.2, make. The event model is a classification model that uses hide and report actions as positive examples, and views without any action as negative examples. The negative experience model also uses hide and report actions as positive examples, but uses like actions as negative examples. Data upsampling for user interactions on content with integrity issues is performed for both models. Furthermore, data from heavy hide users (frequent hides) are downsampled.

We run a no enforcement AB test on Facebook's Connected feed and present the results on FPR, LTR, and DLTR on Table 1. The metrics are organized in ten columns: each column refers to a different prediction threshold; as we reduce the prediction threshold (i.e., the potential demotion would apply to more views), the impacted views percentage increases from 0.5% in the leftmost column to 5%, in the rightmost column. For each model, in each column, we select the prediction threshold that achieves the target impacted views percentage.

For DLTR and LTR, we report ratios relative to Connected Feed averages. For instance, if a model would select views uniformly at random, all DLTR and LTR metrics in this table would be 1.0. For the DLTR metric, the higher the value, the better, while for the FPR and LTR metrics, the lower the value the better.

First, note that as the impacted views percentage increases, from the leftmost to the rightmost column, all metrics get worse for both models, i.e., DLTR decreases and FPR and LTR increase; the only exception is the LTR metrics for the event model, which is already very high at 0.5 impacted views. Hence, as we increase the impact of the models, by targeting more views, the collateral damage in terms of LTR and FPR, inevitably, increases, while the effectiveness, in terms of DLTR, becomes more limited.

By comparing the two models, we can see that the negative experience model has a much smaller collateral damage: both the FPR and LTR are 30 to 100 times lower than the FPR and LTR of the event model, for all the impacted views percentages. The DLTR is 2 to 2.5 times better for the event model (which is more optimized in detecting hides and reports, by design), compared to the negative experience model. Nevertheless, the negative experience model is still better than the Connected feed average DLTR by 30 to 7 times, as the impacted views percentage goes from 0.5% to 5%. In practice, the low collateral damage is very important for a model that targets negative experiences and extends an existing item ranking algorithm.

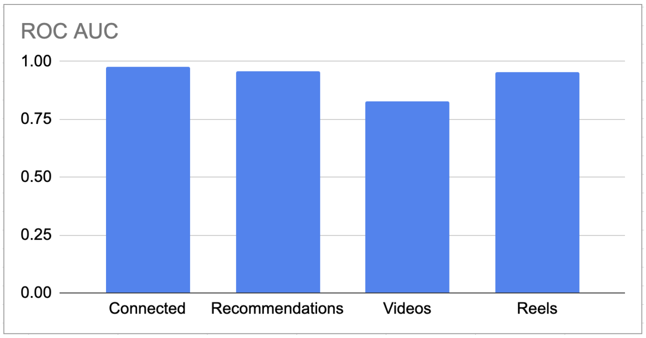

Figure 4: Connected feed model ROC AUC

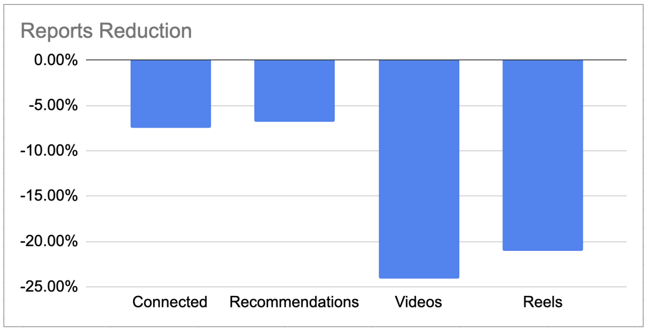

Figure 5: Connected feed reduction in reports

4.2 Impact of negative experience models across the Facebook app

Using four actual enforcement AB tests, we assessed the impact of introducing a negative user experience model in four sections of the Facebook app: Connected Feed, Recommendations, Videos, and Reels. Figure 4 indicates the unweighted ROC AUC for the four models, while Figures 5 to 7 depict the reduction in Reports, Hides, and Feedback submission, by the introduction of the respective model, in each section. The goal here is not to compare the

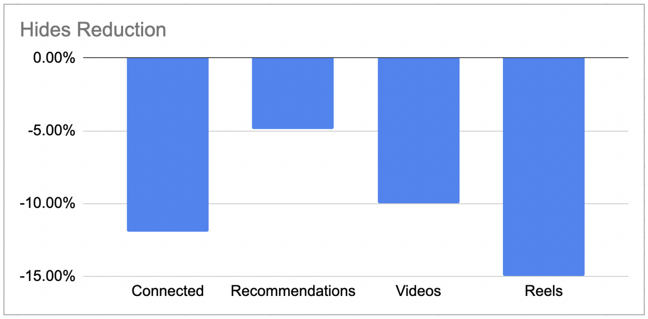

Figure 6: Connected feed reduction in hides

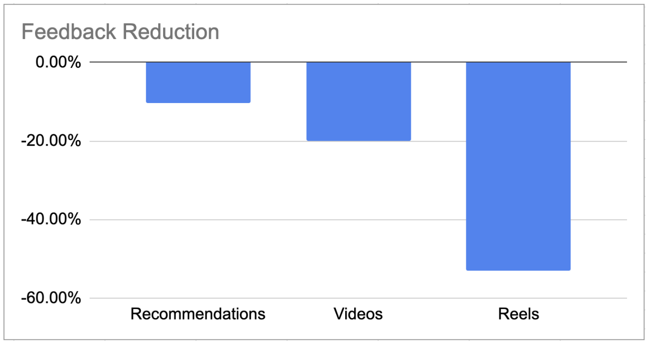

Figure 7: Connected feed reduction in feedback

impact between the different sections, since the sections have many significant differences in the UI and the content or user behaviors.

We have a few very interesting observations:

(1) There are dramatic reductions in reports for video content: higher than 20% for Videos and Reels.
(2) The lower ROC AUC for the Videos model, does not end up in a more limited reduction in hides, feedback, and reports, compared to the other models: this indicates that offline metrics are not proportional to online success and impact.
(3) The impact on Reels indicates the space for improvement in user sentiment, for this more modern social-media type of content. The negative experience model is able to fill this space by bringing reductions of 20%, 15%, 50% on reports, hides, and feedback, respectively.

Overall, the significant reductions in hides, feedback, and reports do not come with any engagement metric regressions and, as we will also see in the next section, they actually impact user sentiment.

4.3 User sentiment survey results

Figure 8 shows the survey results when comparing responses from two groups: treatment users have a Connected feed utilizing a negative experience model, while control users do not have a negative experience model deployed in any app section. We observe a statistically significant improvement in users' sentiment with respect to Nudity, Click Bait, False / Misleading, Hate Witness, and Violence; the improvement ranges from 3% to 6%. The 1.7% improvement in Profanity is on the limit of being statistically significant, while for Ad Farms, Quality, Bullied Witness, we observe improvements, which are not statistically significant however. Given those surveys are not section-specific but generic and refer to the whole Facebook app, achieving statistically significant improvements of that magnitude (3% to 6%) is quite impressive and indicates the impact of even a single-section negative experience model, at an app level.

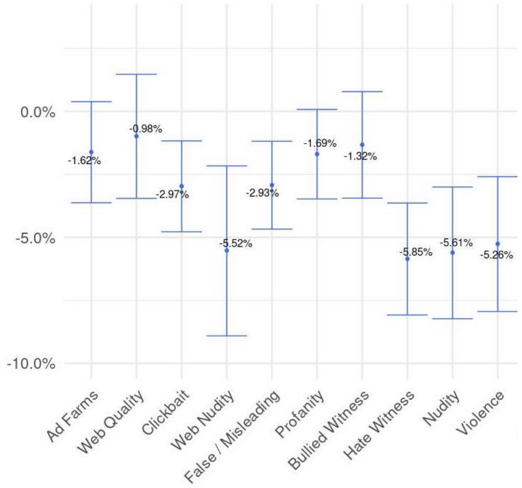

Figure 8: Survey Results

5 RELATED WORK

Our work is closely related to content-based harm detection, user understanding in social media, and personalization models. Detection of negative user experiences in social media is of general interest, with a number of papers analysing negative experience effects [4][11][10]. Our work is motivated by these findings, and aims at detecting and reducing negative experiences in social media platforms. There is also extensive work in the detection of harmful content in social media, related to Misinformation [12], Toxicity [14], Adult and Graphic imagery [2], and others [5]. User understanding in Social Media has also been largely explored [7][8] in the areas of career recommendations [6], online advertising and shopping [17], entertainment [17], and general content recommendations [13] to predict user interests. We believe our work is one of the first attempts to explore the long-term impact of short-term optimized item ranking, in a large scale platform, apply personalization to detect negative experiences, and integrate such a model with item ranking algorithms.

6 CONCLUSION

We studied the problem of improving item ranking long-term effects, in social media platforms, focusing on limiting negative user experiences, without affecting short term engagement goals. The approach we proposed consists of a personalized ML model that predicts a given user's experience when viewing a specific item and a set of integration rules for the item ranking algorithm. In addition, we proposed an evaluation framework consisting of metrics, and AB test and survey specifications. Our experiments via AB testing and surveys showed a 5 to 50 percent reduction in hides, reports, and feedback submissions with user sentiment improvements of 3 to 6 percent on key integrity areas.

REFERENCES

[1] Product Management Anna Stepanov, Director. 2021. Sharing Our Content Distribution Guidelines. Facebook Newsroom. (Sep 2021). https://about.fb.com/news/2021/09/content-distribution-guidelines/

[2] Thomas M Chen. 2021. Automated Content Classification in Social Media Platforms. In *Securing Social Networks in Cyberspace*. CRC Press, 53–71.

[3] Facebook. [n. d.]. Facebook Community Standards. Facebook Transparency Center. ([n. d.]). https://transparency.fb.com/policies/community-standards/?from=https%3A%2F%2Fwww.facebook.com%2Fcommunitystandards%2F

[4] Yany Grégoire, Audrey Salle, and Thomas M. Tripp. 2015. Managing social media crises with your customers: The good, the bad, and the ugly. *Business Horizons* 58, 2 (2015), 173–182. https://doi.org/10.1016/j.bushor.2014.11.001 EMERGING ISSUES IN CRISIS MANAGEMENT.

[5] Alon Y. Halevy, Cristian Canton-Ferrer, Hao Ma, Umut Ozertem, Patrick Pantel, Marzieh Saeidi, Fabrizio Silvestri, and Ves Stoyanov. 2020. Preserving Integrity in Online Social Networks. *CoRR* abs/2009.10311 (2020). arXiv:2009.10311 https://arxiv.org/abs/2009.10311

[6] Rashidul Islam, Kamrun Naher Keya, Ziqian Zeng, Shimei Pan, and James Foulds. 2021. Debiasing Career Recommendations with Neural Fair Collaborative Filtering. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 3779–3790. https://doi.org/10.1145/3442381.3449904

[7] Long Jin, Yang Chen, Tianyi Wang, Pan Hui, and Athanasios V. Vasilakos. 2013. Understanding user behavior in online social networks: a survey. *IEEE Communications Magazine* 51, 9 (2013), 144–150. https://doi.org/10.1109/MCOM.2013.6588663

[8] Sheng Li and Handong Zhao. 2020. A Survey on Representation Learning for User Modeling.. In *IJCAI*. 4997–5003.

[9] Maxim Naumov, Dheevatsa Mudigere, Hao-Jun Michael Shi, Jianyu Huang, Narayanan Sundaraman, Jongsoo Park, Xiaodong Wang, Udit Gupta, Carole-Jean Wu, Alisson G. Azzolini, Dmytro Dzhulgakov, Andrey Mallevich, Ilia Cherniavskii, Yinghai Lu, Raghuraman Krishnamoorthi, Ansha Yu, Volodymyr Kondratenko, Stephanie Pereira, Xianjie Chen, Wenlin Chen, Vijay Rao, Bill Jia, Liang Xiong, and Misha Smelyanskiy. 2019. Deep Learning Recommendation Model for Personalization and Recommendation Systems. *CoRR* abs/1906.00091 (2019). arXiv:1906.00091 http://arxiv.org/abs/1906.00091

[10] Michelle O'Reilly. 2020. Social media and adolescent mental health: the good, the bad and the ugly. *Journal of Mental Health* 29, 2 (2020), 200–206. https://doi.org/10.1080/09638237.2020.1714007 arXiv:https://doi.org/10.1080/09638237.2020.1714007 PMID: 31989847.

[11] Michelle O'Reilly, Nisha Dogra, Natasha Whiteman, Jason Hughes, Seyda Eruyar, and Paul Reilly. 2018. Is social media bad for mental health and wellbeing? Exploring the perspectives of adolescents. *Clinical Child Psychology and Psychiatry* 23, 4 (2018), 601–613. https://doi.org/10.1177/1359104518775154 arXiv:https://doi.org/10.1177/1359104518775154 PMID: 29781314.

[12] Kellin Pelrine, Jacob Danovitch, and Reihaneh Rabbany. 2021. The Surprising Performance of Simple Baselines for Misinformation Detection. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 3432–3441. https://doi.org/10.1145/3442381.3450111

[13] Karthik Raja Kalaiselvi Bhaskar, Deepa Kundur, and Yuri Lawryshyn. 2020. Implicit Feedback Deep Collaborative Filtering Product Recommendation System. *arXiv e-prints* (2020), arXiv–2009.

[14] Martin Saveski, Brandon Roy, and Deb Roy. 2021. The Structure of Toxic Conversations on Twitter. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 1086–1097. https://doi.org/10.1145/3442381.3449861

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[16] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-Attention with Linear Complexity. https://doi.org/10.48550/ARXIV.2006.04768

[17] Haitian Zheng, Kefei Wu, Jong-Hwi Park, Wei Zhu, and Jiebo Luo. 2020. Personalized fashion recommendation from personal social media data: An item-to-set metric learning approach. *arXiv preprint arXiv:2005.12439* (2020).