Amplification Pipelines

The Role of Feedback Loops in Recommender System Bias

Candidate: Caio Truzzi Lente
Advisor: Prof. Dr. Roberto Hirata Jr.

DCC-IME-USP | 2023-05-24

Motivation

- Social networks are ubiquitous: socializing, reading news, expressing ourselves
- The public wants to know what role their platforms might have in radicalizing users, specially younger ones
  - Mainly anecdotal evidence (e.g. Facebook depression experiments, YouTube's bizarre videos aimed at kids, etc.)
- Journalists and specialists alike argue that social media's algorithms are tuned to peddle conspiracy theories, extremist views, and false information
- The debate around the role of recommender systems in social media radicalization is still too recent and based in anecdotes
- More quality research is vital to inform both the public and opinion makers about if and how much recommendation algorithms influence social media users

Methods

- Recommender systems: providing users with personalized product or service recommendations
    - Trade secrets, but known to gather enormous amounts of data about the user's interaction with the website
    - Algorithms might have **explicit biases**: YouTube's system, for example, explicitly favors more recent videos
    - Algorithm might develop **implicit biases**: Instagram's system, for example, learned its user's differentiated homophily and favored male profiles
- Goal: understand the mechanisms through which recommender systems can end up learning or developing biases (which might lead to radicalization)
    - Study how and how fast recommender systems develop biases and whether this can create **amplification pipelines**

Literature review

- A.-A. Stoica et al. (2018). *Algorithmic Glass Ceiling in Social Networks: The effects of social recommendations on network diversity*
- M. Ledwich et al. (2019). *Algorithmic Extremism: Examining YouTube's Rabbit Hole of Radicalization*
- R. Jiang et al. (2019). *Degenerate feedback loops in recommender systems*
- Z. Zhao et al. (2019). *Recommending what video to watch next: a multitask ranking system*
- M. H. Ribeiro et al. (2020). *Auditing radicalization pathways on YouTube*
- S. Yao et al. (2021). *Measuring Recommender System Effects with Simulated Users*
- Y. Li et al. (2022). *Fairness in Recommendation*

Proposal

- **Static analysis**: doesn't take into account the evolution of the system after multiple rounds of training and learning from new data
    - Hypothesis: even a simple recommendation algorithm can demonstrate some sort of bias towards a subset of of items
    - Given an algorithm that is user agnostic, would the resulting recommender system still favor any items?
- **Dynamic analysis**: takes into account the dynamics of the system, i.e., the algorithm learning for the users' feedbacks to its recommendations
    - Hypothesis: if the users reinforce the beliefs of the algorithm it will degenerate and only recommend a subset of items
    - How fast does a degenerate feedback loop develop, ignoring personal preferences and distinctions between films?

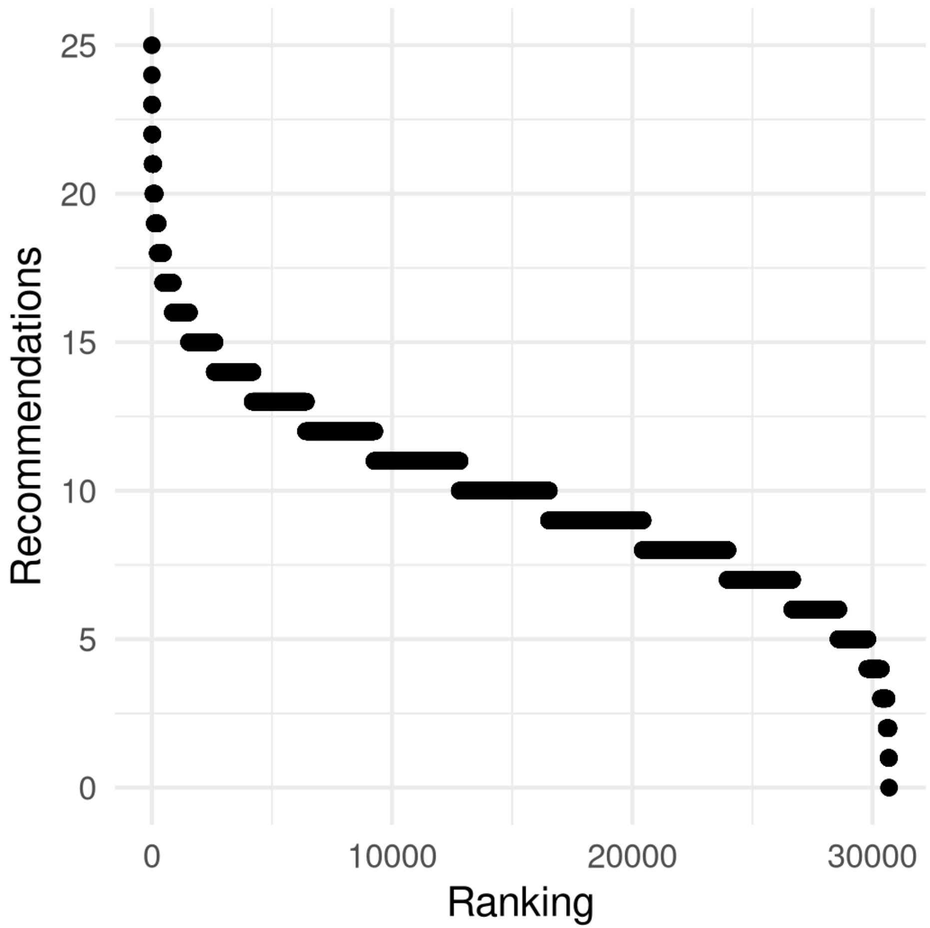
Datasets

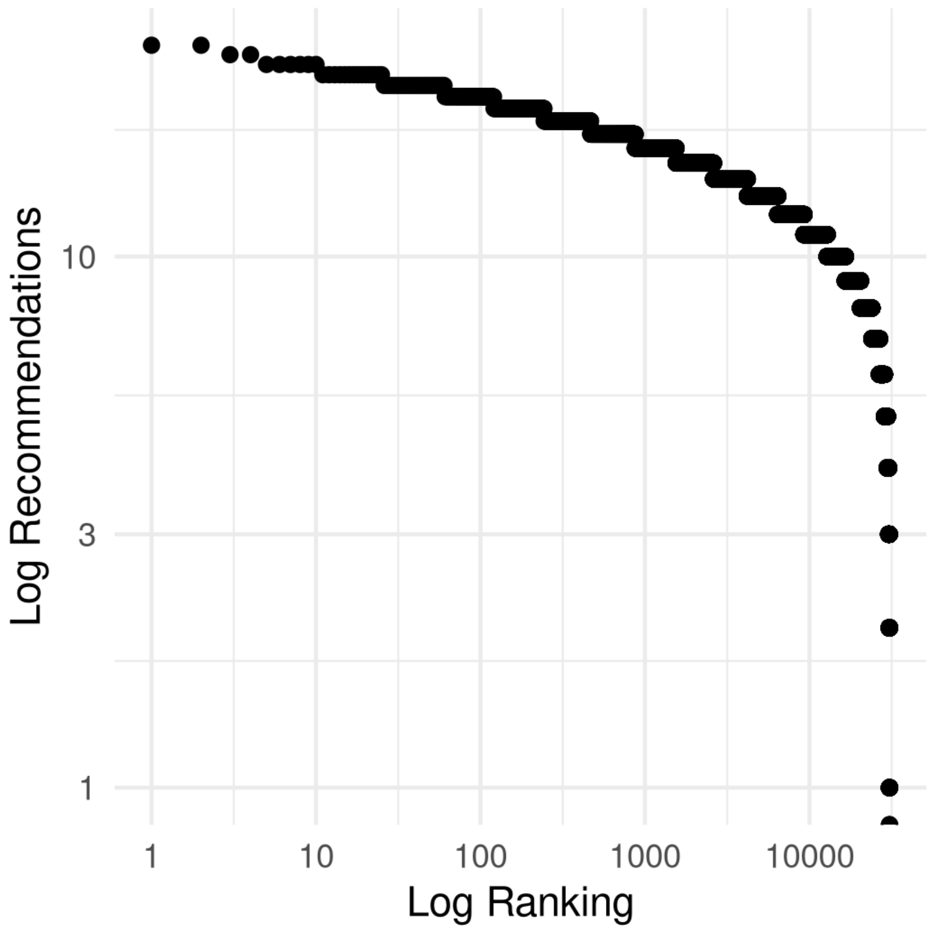
- The main dataset used for experimentation was MovieLens (Harper et al., 2015), a dataset about movie ratings
  - 25M ratings applied to 62K movies by 162K users, enriched with information about the movies' credits, metadata, keywords, and links
  - A sample of 30,689 movies was taken in order to reduce the hardware requirements of iterative experimentation
- The dataset used to validate hypotheses was Book-Crossing (Ziegler, 2004), a dataset about book reviews
  - 1.1M ratings applied by 278K users to 271K books, and information like title, author, publisher, etc.
  - A sample of 20,000 books was taken in order to reduce the hardware requirements of iterative experimentation

Static analysis

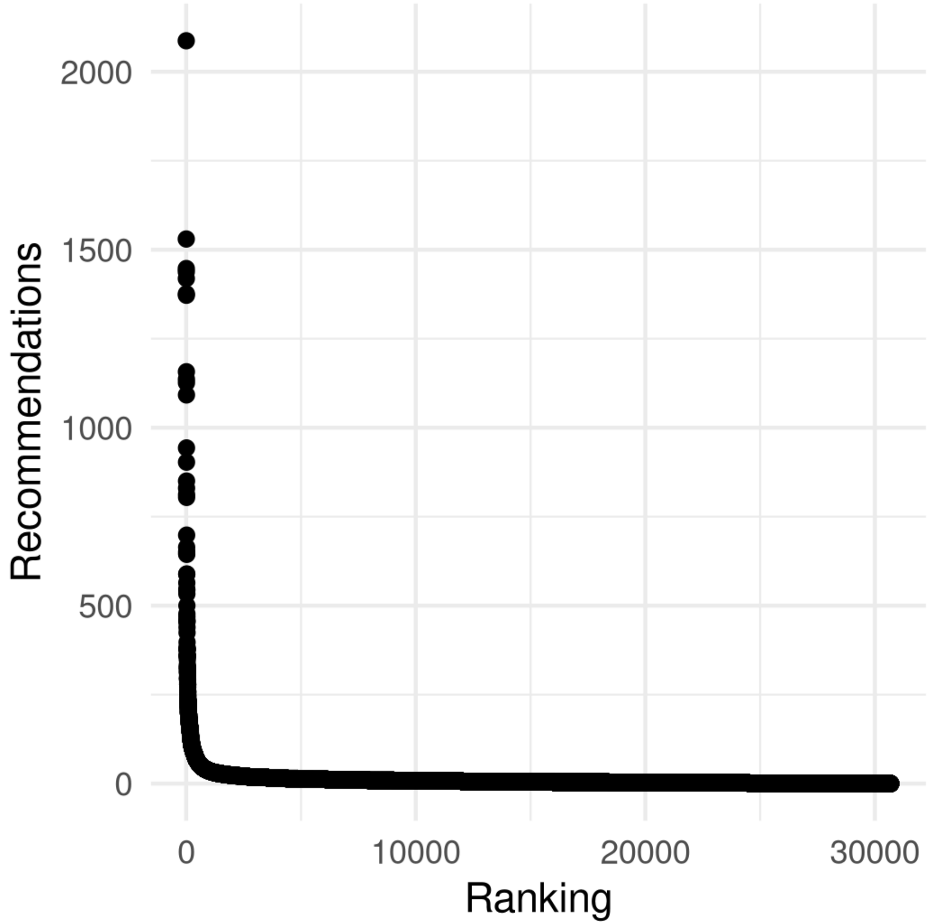
- Excluding user information is important because they might transfer their own biases to the model
- "Recommendation profiles": a summary of how many times an arbitrary item is recommended overall
  - Trivial model: a simple sampler that returns n movies at random
  - Vanilla model: cosine similarity applied to vector representations of the items
  - Cutoff models: uses cutoff points after which words would not be included in the vector representations
  - Similarity models: uses other distance metrics (cosine distance, Euclidean distance and Manhattan distance)
  - Vanilla model with synthetic metadata: the sparsity of the vector representations are controlled by how many elements should be non-zero

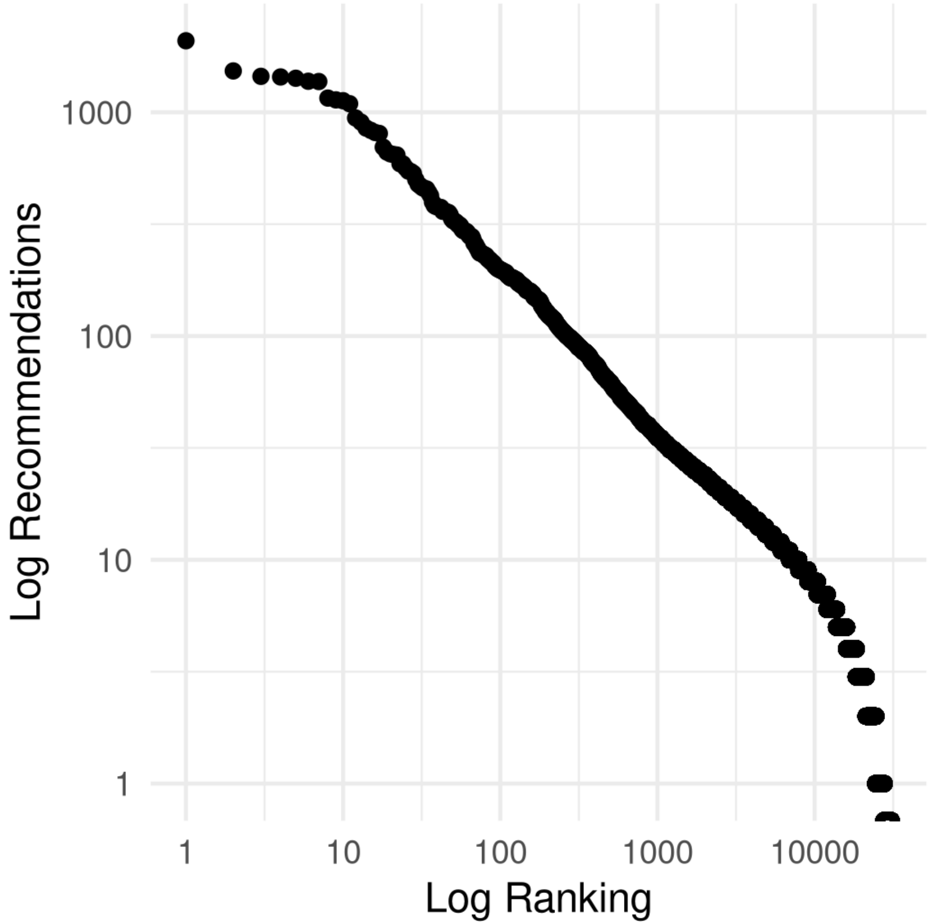
Trivial model



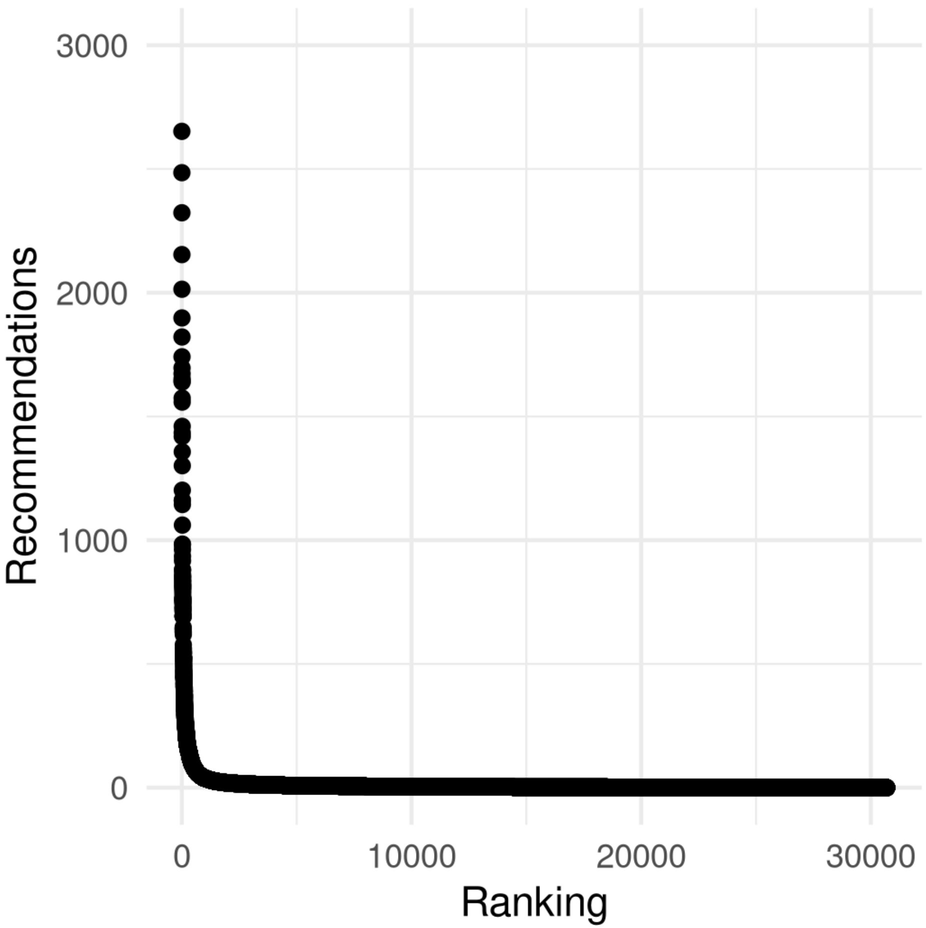


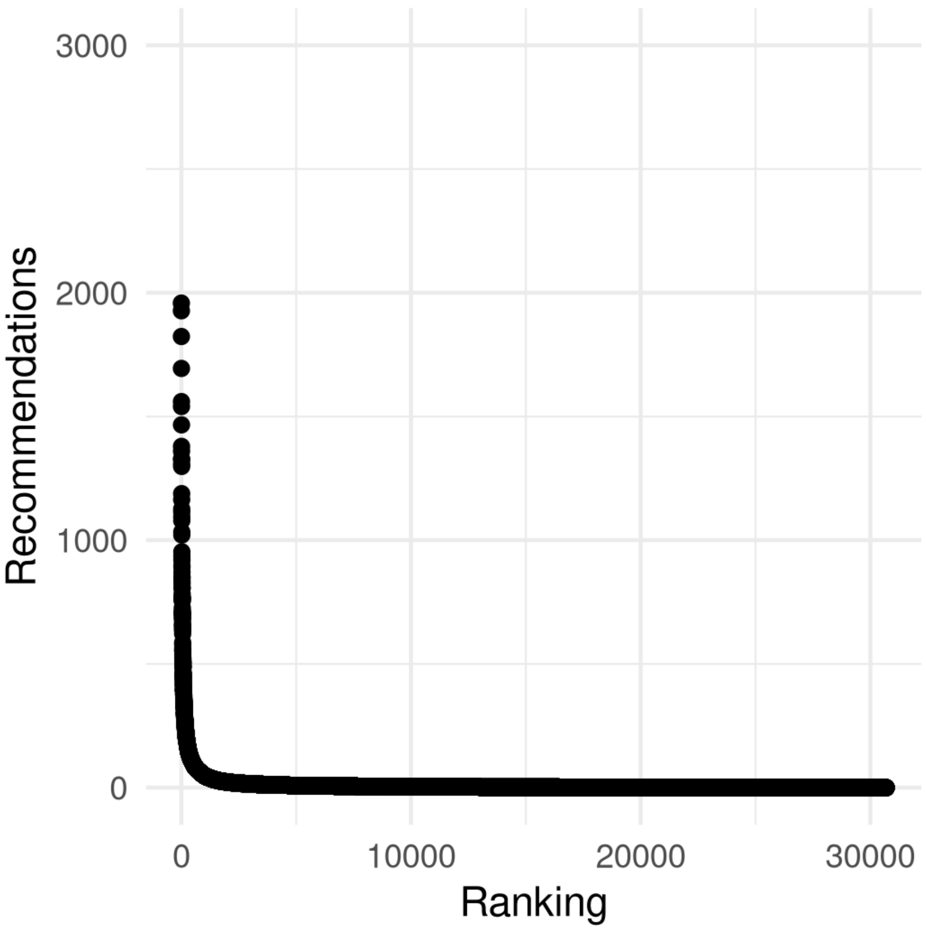
Vanilla model



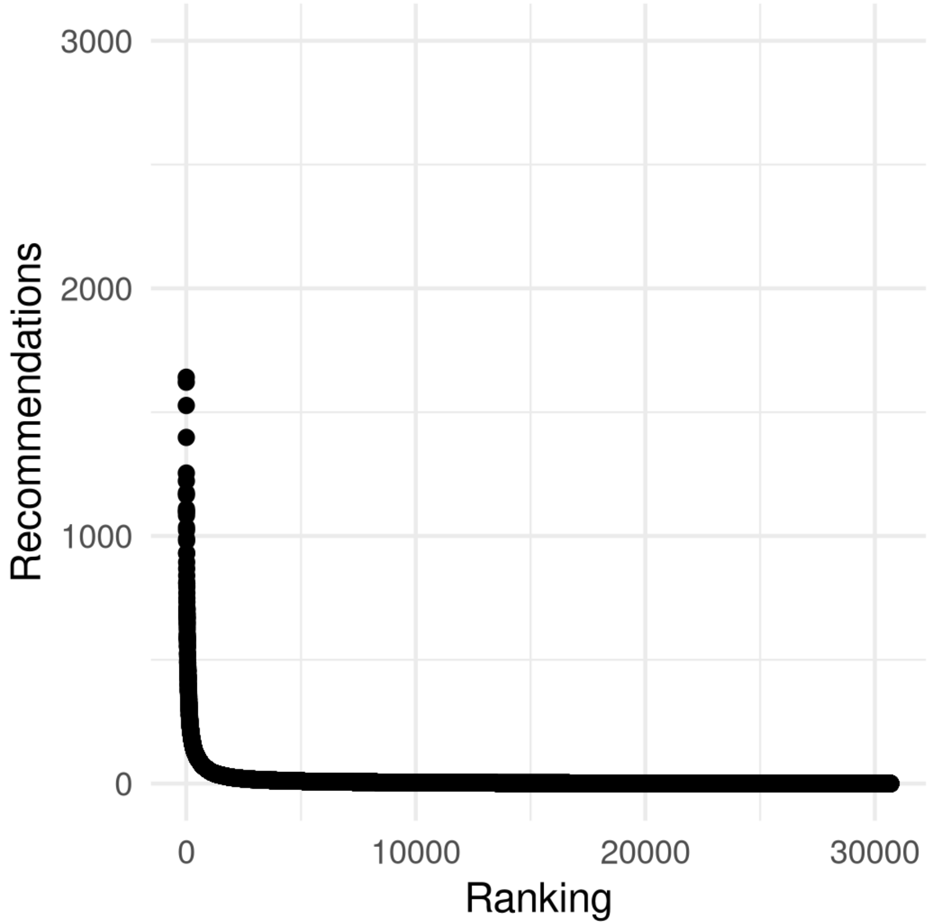


Cutoff models

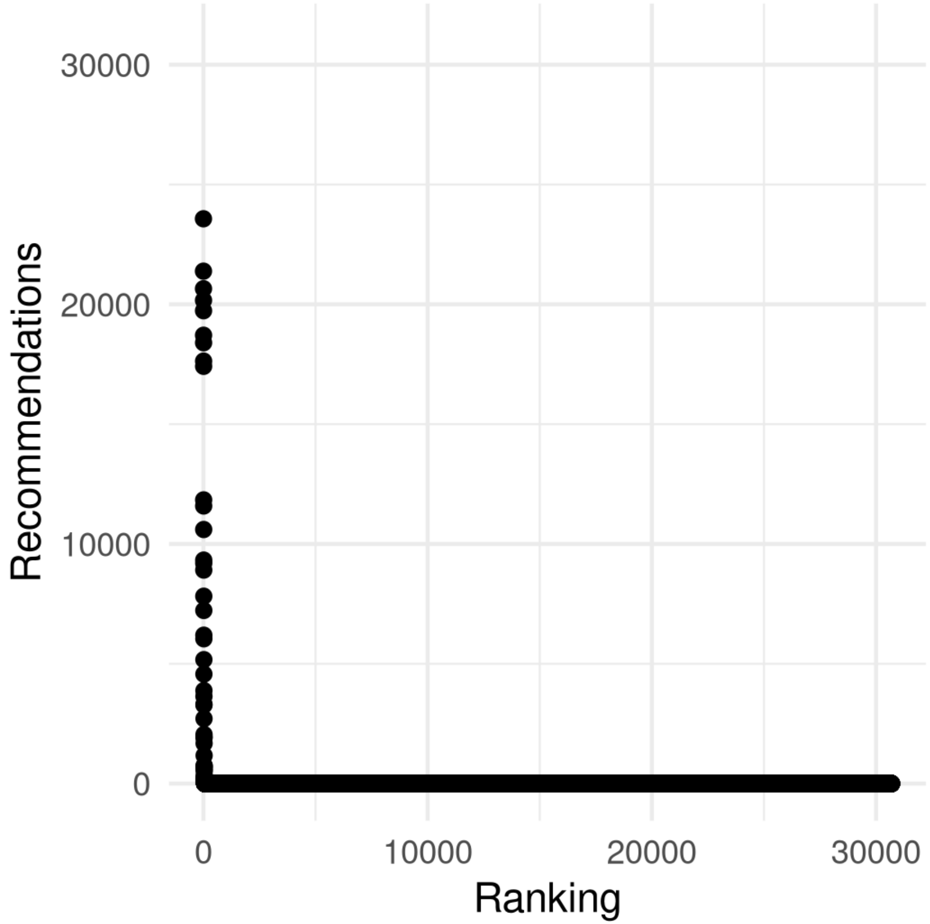


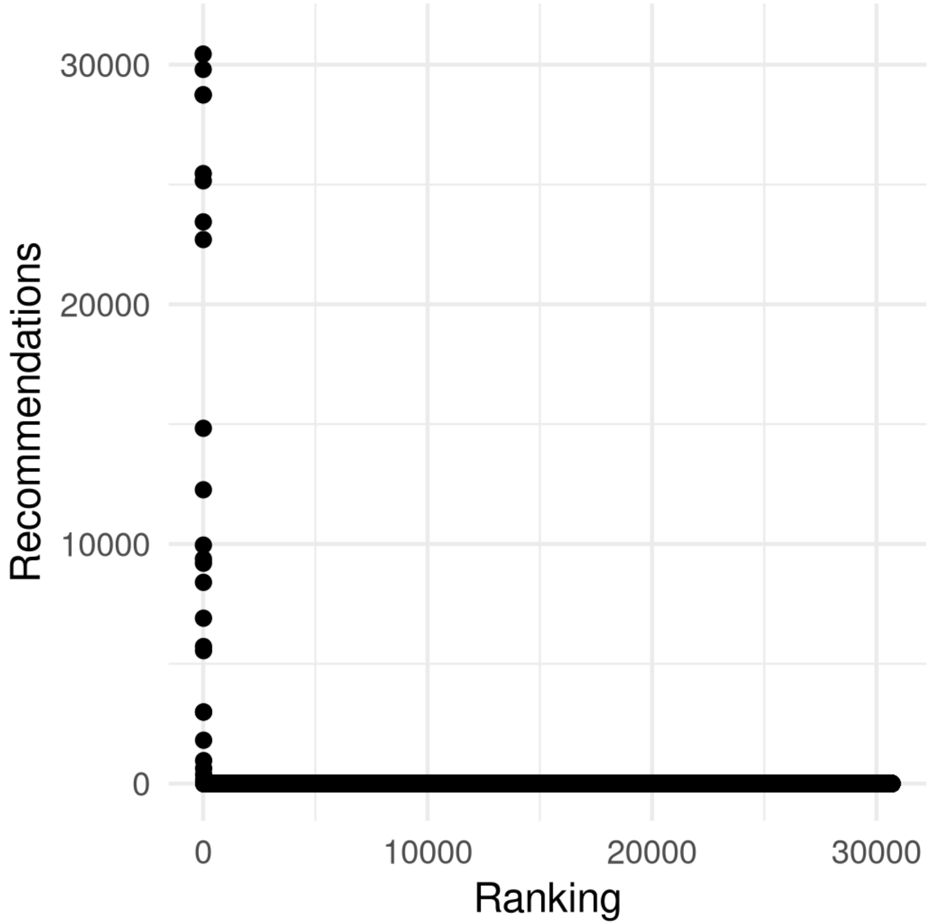


Cutoff model (cont.)

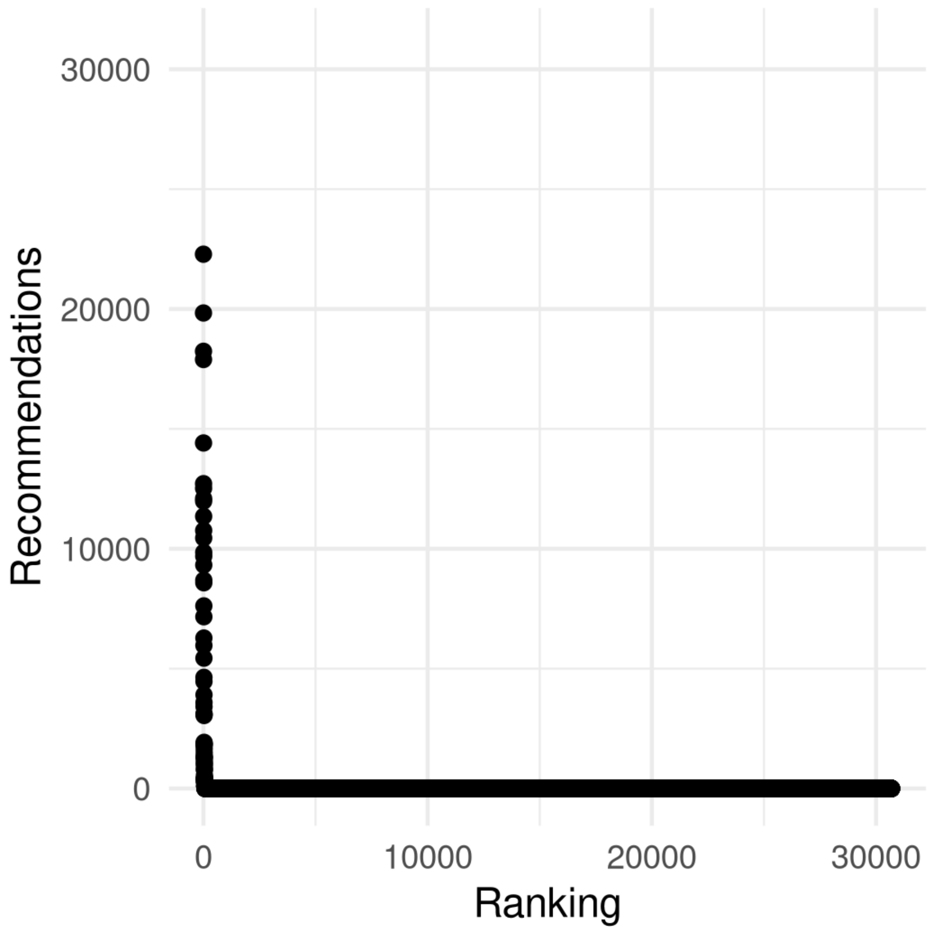


Similarity models

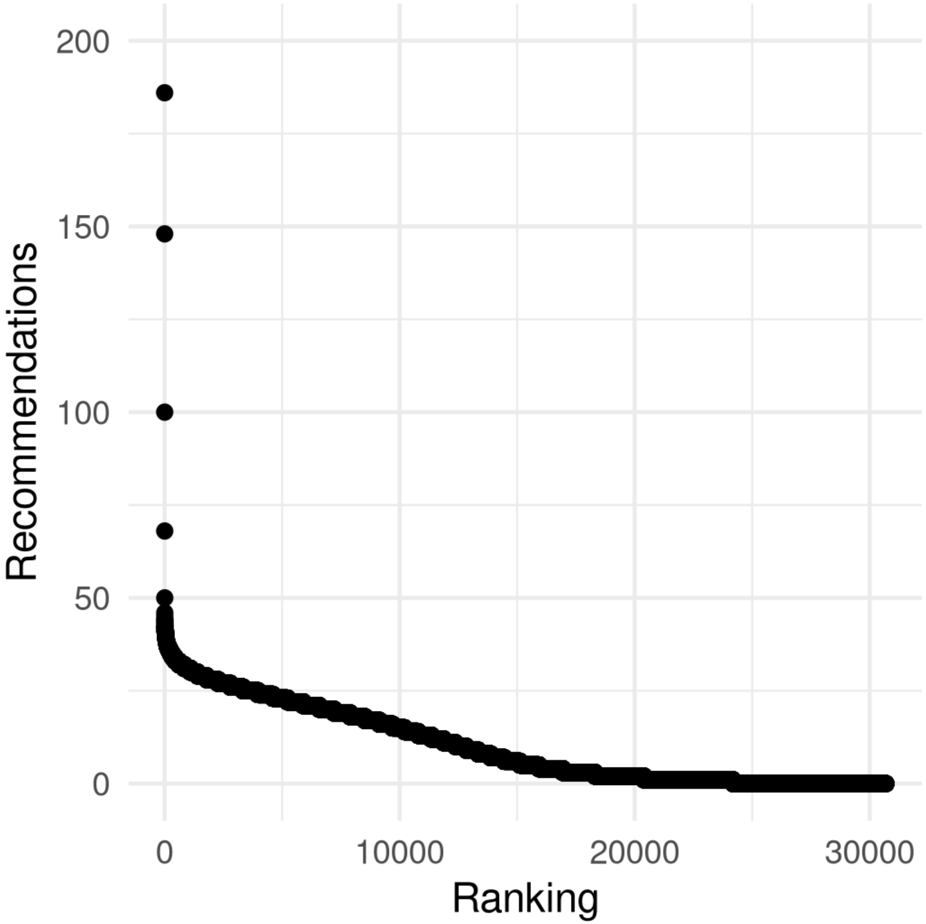


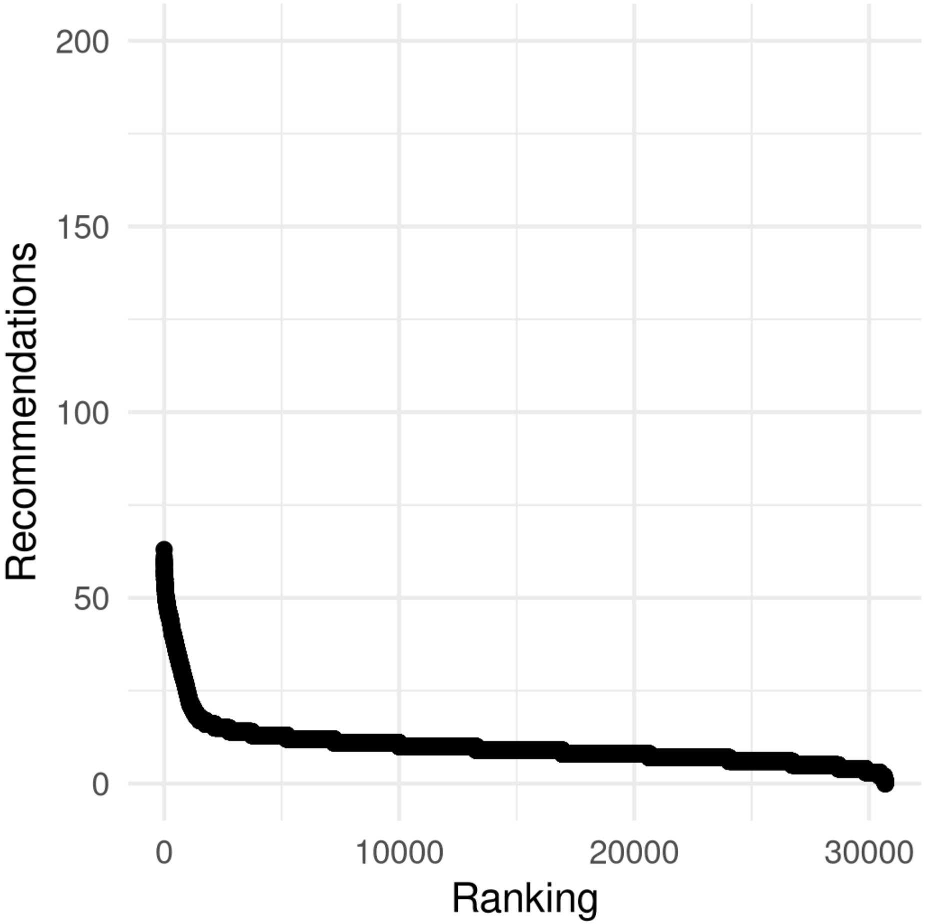


Similarity models (cont.)

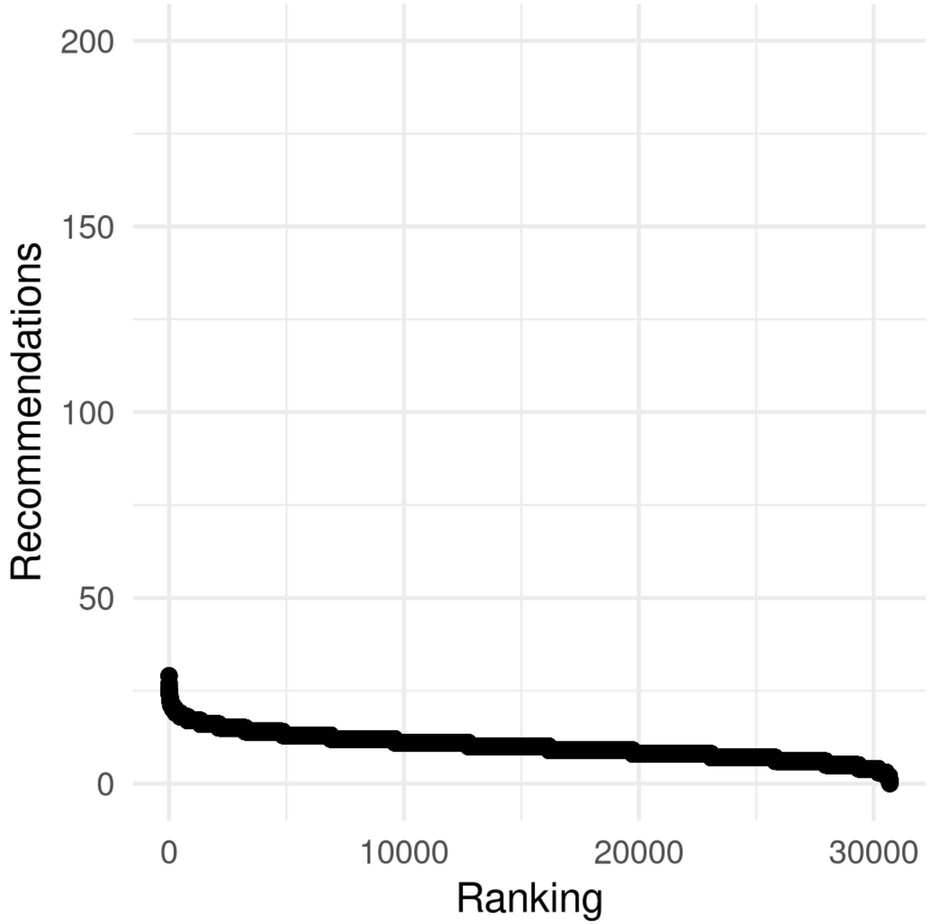


Synthetic metadata





Synthetic metadata (cont.)



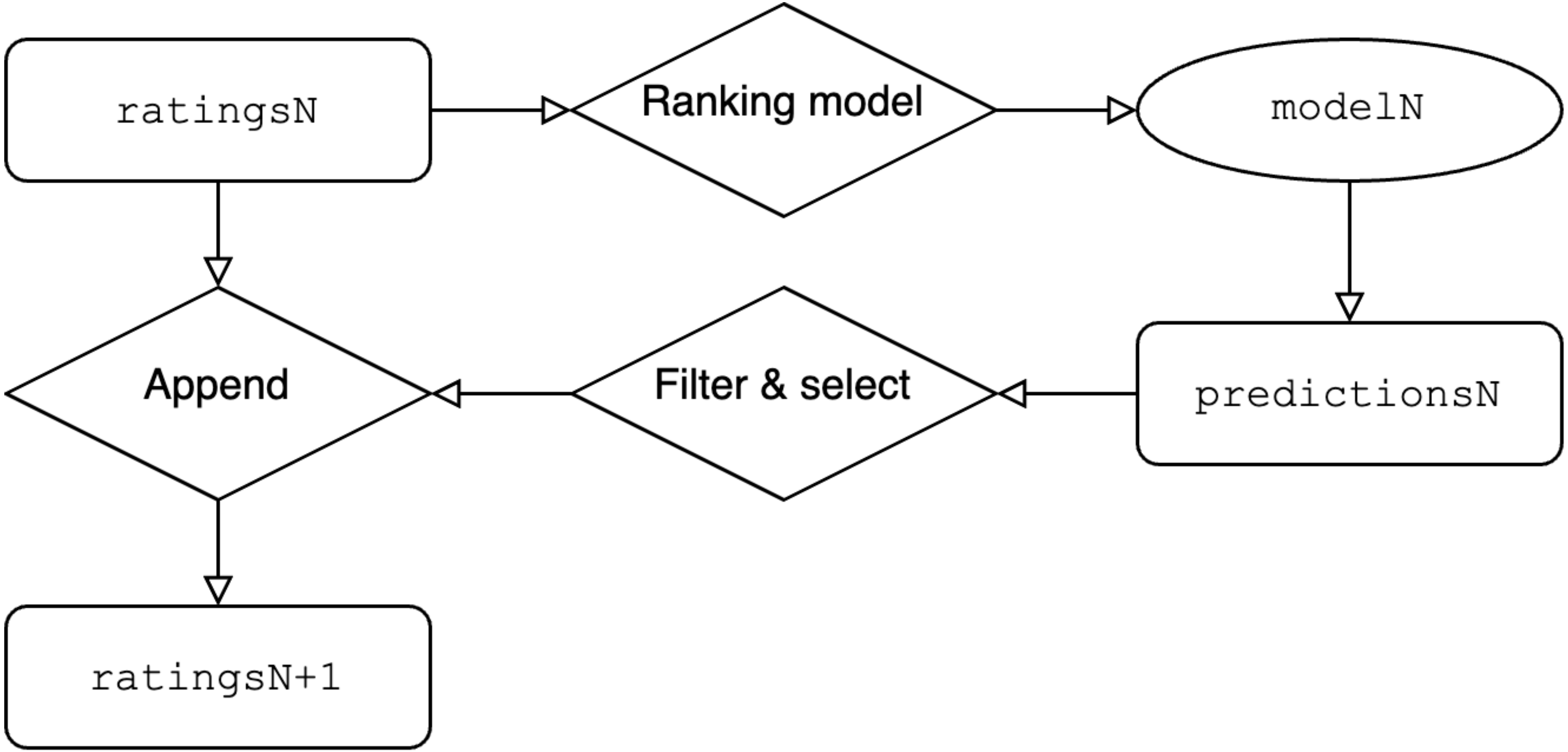
Dynamic analysis

- Deep learning TensorFlow model that is trained for five interactions, always receiving positive feedback from the simulated users
- Better understand the recommendation profiles by modeling it with standard statistical models
  - Exploratory analysis: recommendation entropy, recommendation profile over time, item log-popularity over time
  - Poisson and negative binomial distributions chosen to model number of recommendations over five interactions
  - Simulated envelopes to assess global goodness-of-fit: fits the model to each simulated response variable and obtains the same model diagnostics

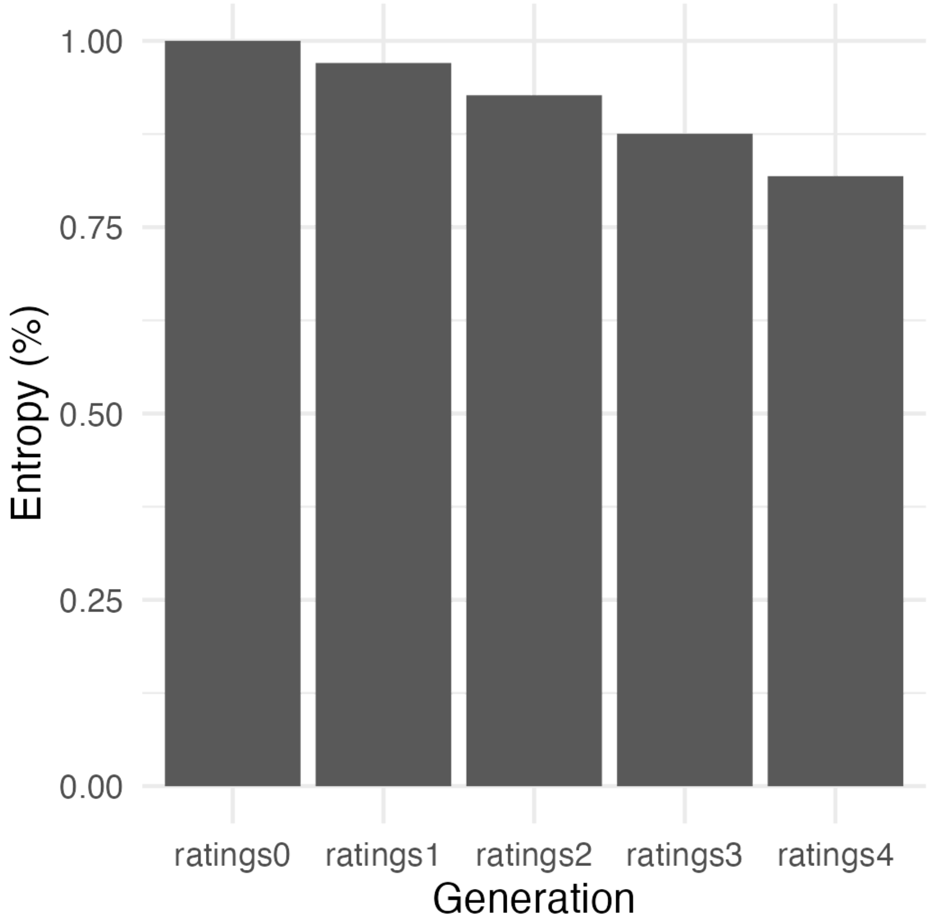
Dynamic analysis (cont.)

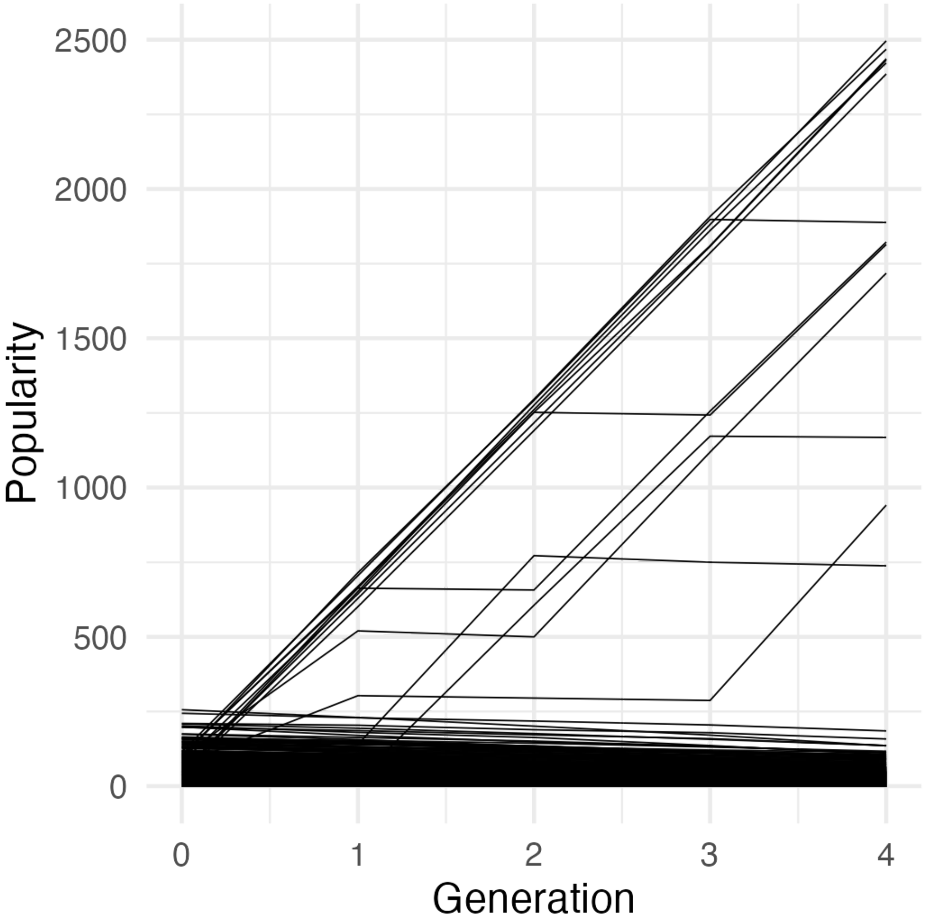
- $\log(\lambda_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$, with popularity being modeled by the factor crossing of generation, genre and average rating.
  - Poisson regression: uses R's `glm()`, simulated envelope uses the `hnp` package
  - Negative binomial regression: uses `MASS`'s `glm.nb()`, simulated envelope uses the `hnp` package
- $\log(\lambda_i) = \delta + \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$, where $\delta \sim \mathcal{N}(0, \psi)$ allows us to better model longitudinal observations and serially correlated errors
  - Mixed-effects Poisson regression: uses the `glmmTMB` package, simulated envelope uses the `DHARMa` package
  - Mixed-effects negative binomial regression: same as previous model

Experiment pipeline

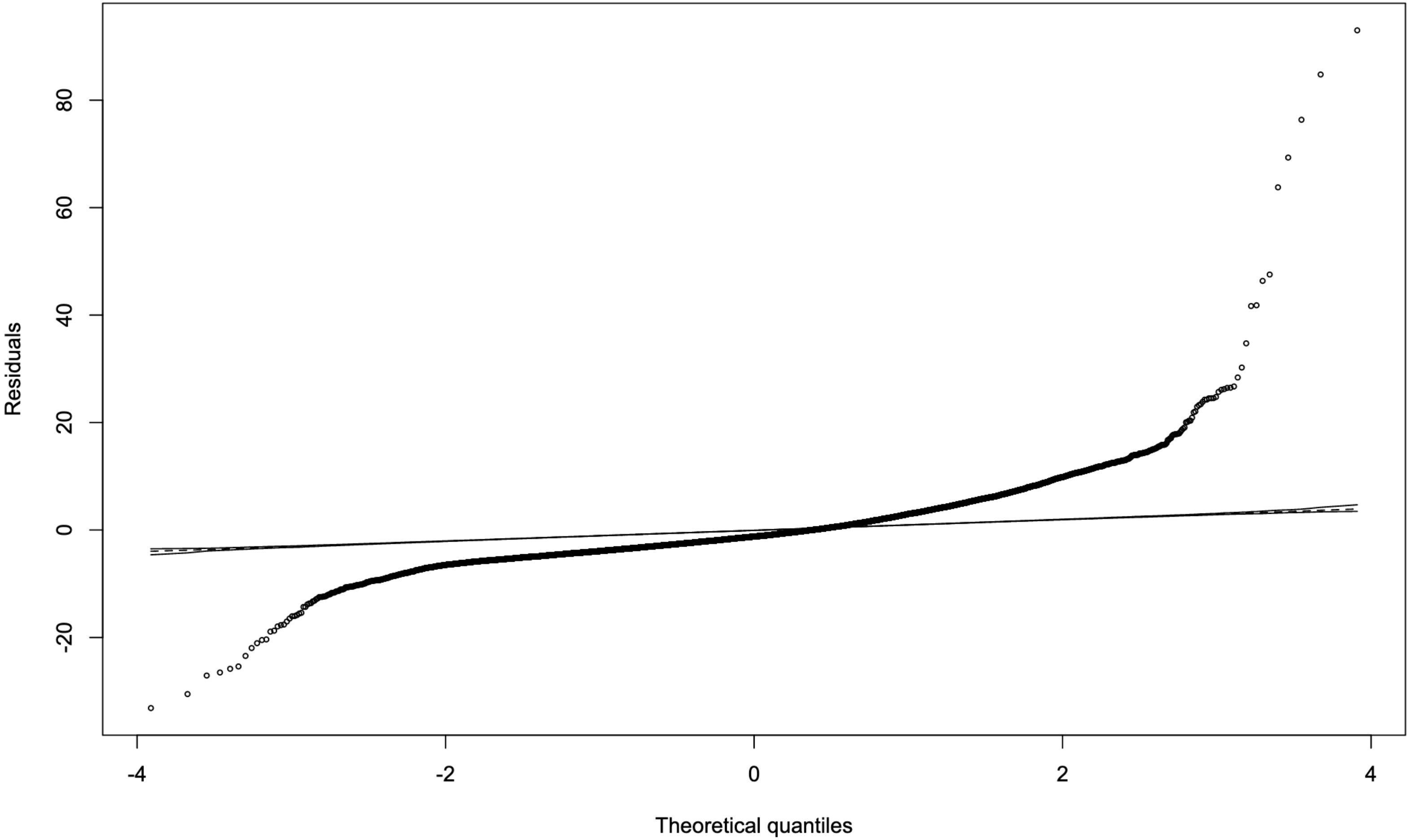


Exploratory analysis

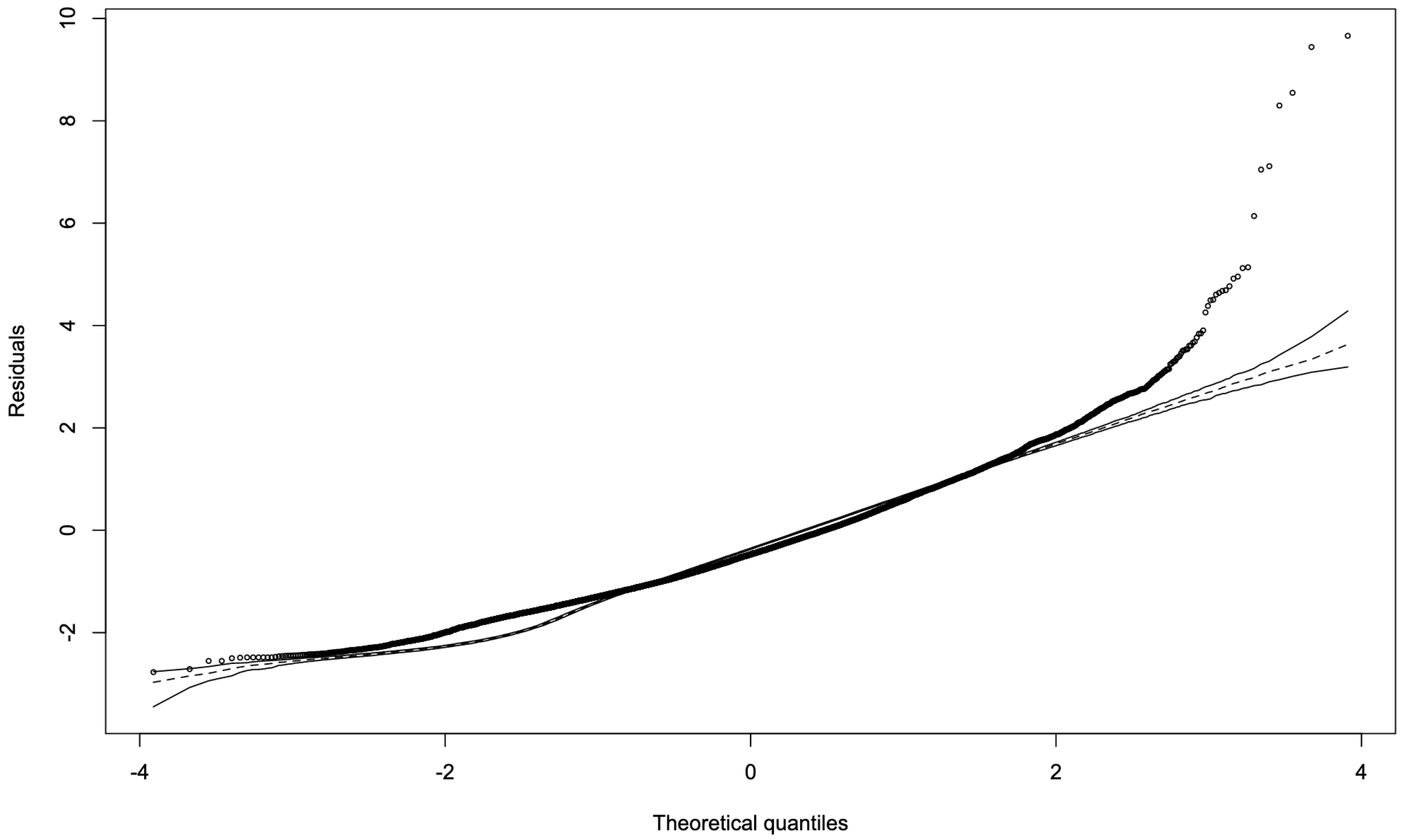




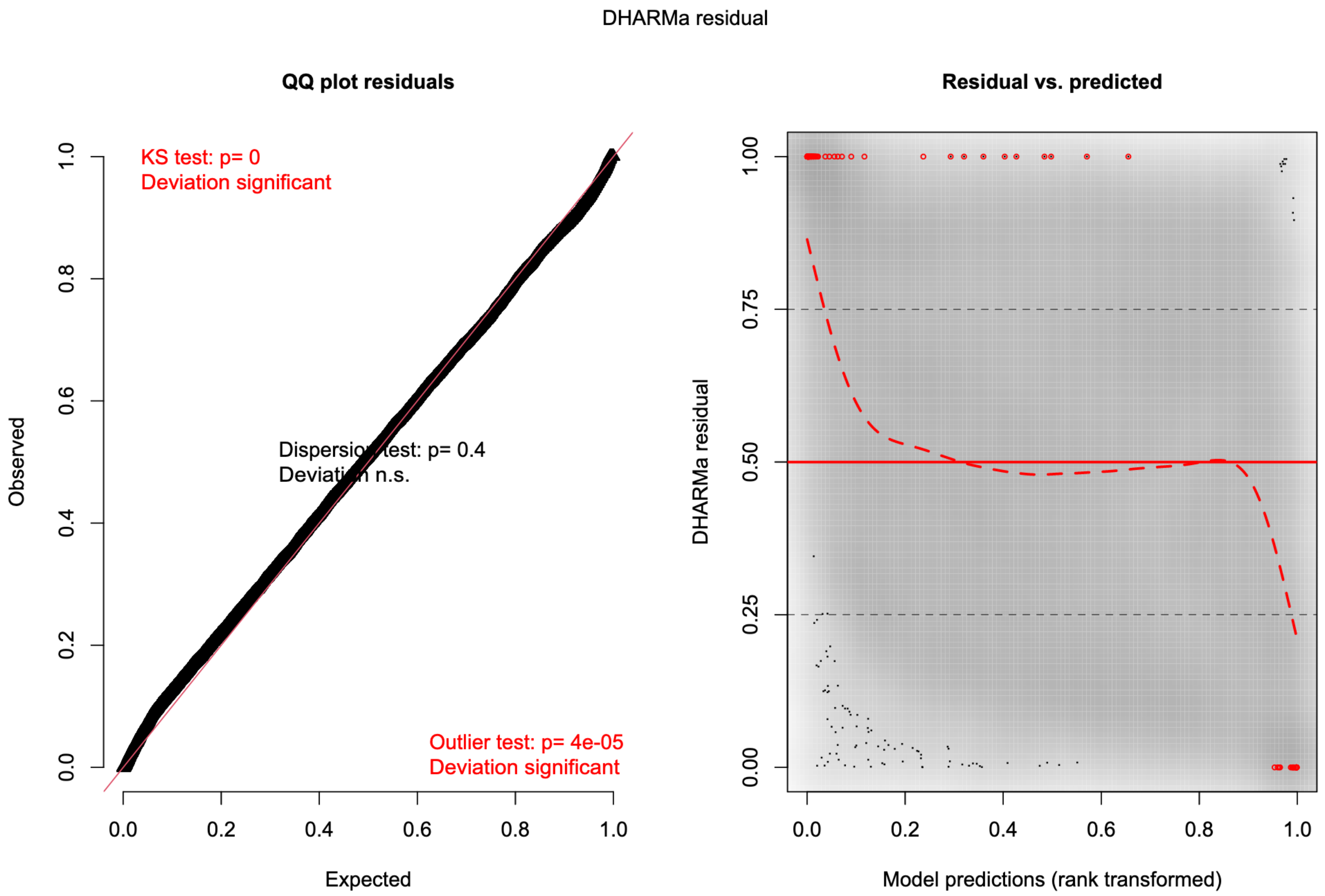
Poisson regression



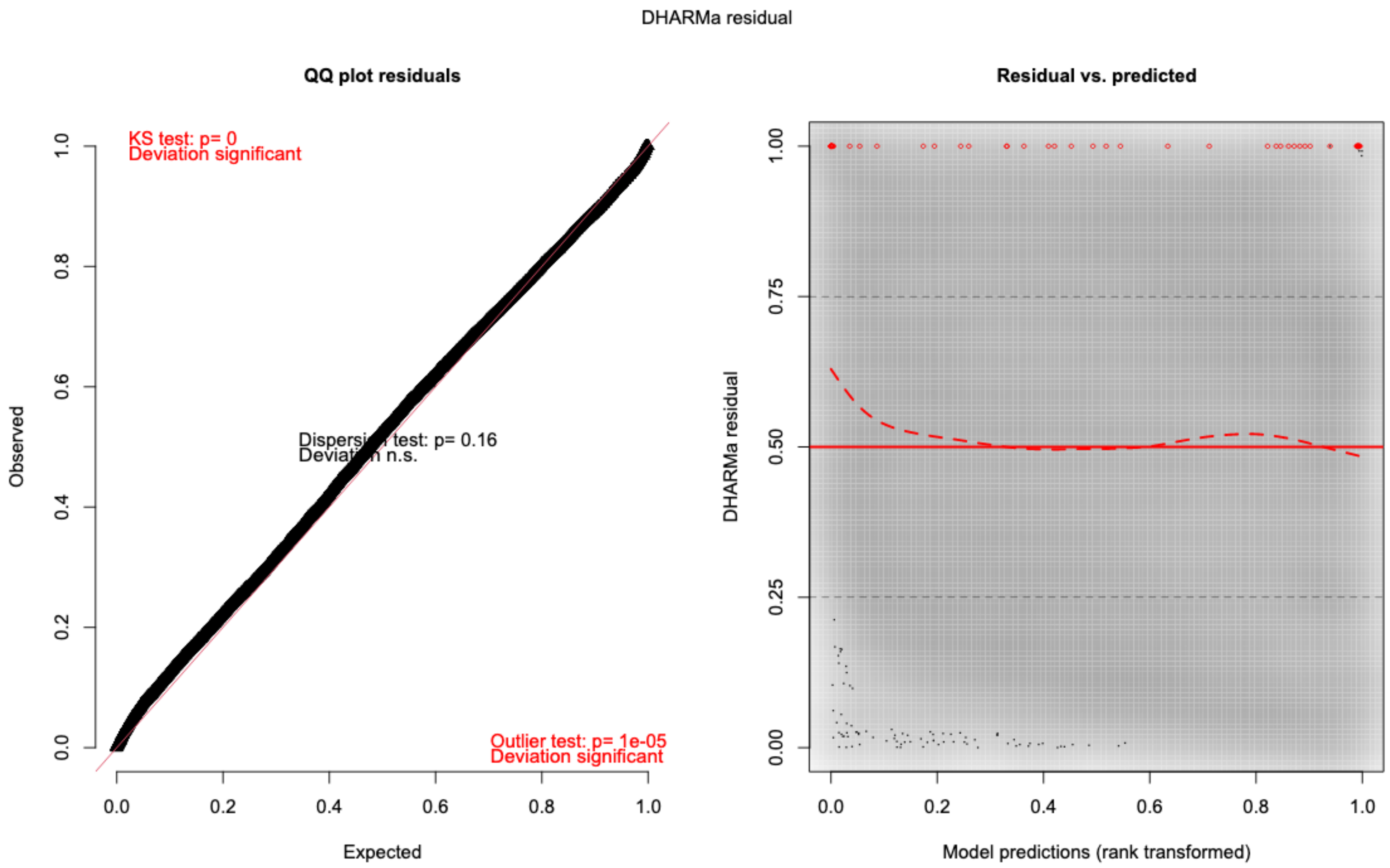
Negative binomial regression



Mixed-effects Poisson regression



Mixed-effects negative binomial regression



Conclusion

- Static analysis
  - Expectation: a systems's recommendation profile would maintain the original dataset's item popularity distribution
  - Reality: during our experiments, all of the tested models displayed exponential or super-exponential recommendation profiles
- Dynamic analysis
  - Expectation: the systems's recommendation profile would grow steeper over time following a well-behaved statistical distribution
  - Reality: the recommendation profiles of our model were tending quickly towards a degenerate distribution
- "Recommender systems, if left unchecked, tent towards a confinement dynamic where points of view [...] are amplified by a system that has to learn from itself."

Future works

- Recent developments: new fairer recommender systems or new metrics, other possible mechanisms through which users can become radicalized
- Further research is still needed in order to find an unambiguous causal link between degenerate feedback loops and user radicalization
  - A possible next step would involve applying the methods discussed in this work to more complex recommendation systems and to larger real-world datasets
  - Another possible improvement would be to simulate users that ignore their recommendations
- We plan on writing and publishing a paper detailing our findings, possibly adding larger-scale experiments

Thank you