# Anonymization and Subsequent Deanonymization of Letterboxd Ratings Data

Jordan Bass
Northwestern University
jordanbass2024@u.northwestern.edu

Nicket Mauskar
Northwestern University
nicketmauskar2024@u.northwestern.edu

Albert Lu
Northwestern University
albertlu2024@u.northwestern.edu

Elijah Hansen
Northwestern University
elijahhansen2024@u.northwestern.edu

## ABSTRACT

The anonymization and privacy of microdata, specifically information on individuals, preference, recommendations, and more, alludes to a wider conversation of the balance and trade-off between privacy and utility. In the context of microdata, which refers to data on individuals or small groups, the privacy-utility trade-off and anonymization become particularly relevant as it becomes easier to re-identify individuals from seemingly anonymous data, which can threaten privacy. At the same time, removing too much identifying information can render the data less useful for analysis and decision-making.

Thus, in order to navigate this trade-off, we sought to find the most effective techniques to anonymize a dataset and then subsequently test an algorithm to deanonymize the same dataset. Our approach involved utilizing a primary dataset from LetterBoxd and supplementing it with an auxiliary dataset from MovieLens, both containing user data, movie information, and ratings. In terms of anonymization, Data Generalization ended up being the most effective technique of the few we chose. In terms of deanonymization, we found that when implementing the Scoreboard_RH function, developed by Arvind Narayanan and Vitaly Shmatikov, we were able to find a few potential matches across the datasets with high certainty, showing proof of concept in the effectiveness of the algorithm.

## 1 INTRODUCTION

The anonymization and privacy of microdata, specifically information on individual preference, recommendations, and ratings, is an important topic because in the modern world, there is a constant battle between privacy and utility. On one hand, there is the usefulness that comes from the sharing of personal information, while on the other hand, there are loads of privacy and anonymity concerns that must be addressed. In 1997, the state of Massachusetts released a dataset of anonymized medical records for employees that worked for the state as well as their dependents. In an effort to deanonymize the data, the state believed that removing personal identifiers such as names and addresses would be sufficient. However, in 2002, a researcher named Latanya Sweeney was able to re-identify several individuals in the dataset, including then-governor William Weld.[1] Sweeney was able to do this by matching the de-identified data to publicly available data, such as voter registration lists and other publicly available information at the time. Sweeney was able to link the medical records to individuals' identities, showing that the dataset was not truly anonymous. Furthermore, through her research on this Massachusetts dataset, Sweeney introduced a model called $k$-anonymity, which remains an important metric to determine the privacy of datasets to this day (and will be used in this paper to demonstrate anonymization effectiveness). This incident highlighted the importance of strong privacy protections for sensitive data and the potential risks of re-identification, even when personal identifiers are removed.

More recently, in 2006, Netflix released their "Netflix Prize" dataset as a part of a competition to improve their recommendation algorithm for users. Under the assumption that Netflix had anonymized their dataset enough, a group of researchers, Arvind Narayanan and Vitaly Shmatiko, were able to successfully develop various deanonymization techniques and subsequently applied one, the scoreboard method, to the Netflix prize dataset in their research titled "Robust De-anonymization of Large Sparse Datasets".[2] They were able to successfully reveal the identities of 99% of users using only eight movie ratings (of which two could be incorrect) and dates of the review (with up to a fourteen day error). The deanonymization of the Netflix Prize dataset, which will be described in detail throughout this paper, is a perfect example of how irresponsible sharing of microdata can lead to a breach of privacy to which a user is entitled, according to most privacy policies. Microdata such as user reviews or ratings may not seem like a breach of privacy in the moment, but if an attacker knows a user's preferences, they could use that information in the future, which breaches the user's privacy. Naryanan and Shmatikov's work is the inspiration for our project, as we aim to accomplish the same goal as them on a different dataset.

We hope to draw conclusions that link the hyperspecificity of modern online usage to possible privacy issues, such as unwanted identification. In particular, we aim to identify specific users that reviewed movies both on LetterBoxd and MovieLens. As the previously mentioned researchers did, if we can reveal the user's preference, that would constitute a significant privacy breach in the user's future. To accomplish this, we plan on first anonymizing a LetterBoxd dataset of movie ratings with user masking, data generalization, swapping, removal, and adding synthetic data, then utilizing the same deanonymization method as Narayanan and Shmatikov to

[1] Sweeney, L. (2002). k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5), 557-570.

[2] Narayanan, A., Shmatikov, V. (2008). Robust De-anonymization of Large Sparse Datasets. Proceedings of the IEEE Symposium on Security and Privacy, 111-125.

validate the effectiveness of the anonymization methods using measurements such as $k$-anonymity. Results from this experiment reinforce the importance of data privacy and anonymization in all data that relates to user preferences such as online shopping reviews, multimedia ratings and reviews, and venue reviews on sites like Yelp.

## 2 BACKGROUND/RELATED WORK

Narayanan and Shmatikov's paper "Robust De-anonymization of Large Sparse Datasets" demonstrated the vulnerabilities of anonymization techniques to deanonymization attacks.[3] In their research, they showed that individuals can be re-identified from a large, sparse dataset, even when their personally identifying information has been removed or replaced with pseudonyms. To conduct their research, Narayanan and Shmatikov used the Netflix Prize dataset, which was publicly released information. Netflix anonymized the data by removing personally identifying information, such as names and addresses, and assigned each user a unique identifier. Despite these measures, Narayanan and Shmatikov were able to match some users to their real identities using auxiliary data sources, such as IMDB ratings and social media profiles.

Narayanan and Shmatikov developed a re-identification algorithm where an attacker has knowledge of additional data about the users that is not present in the anonymized dataset. The deanonymization algorithm that we replicated for use is their Scoreboard_RH algorithm, which involves creating a "scoreboard" of possible matches between users in the anonymized dataset and users in the auxiliary source based on shared attributes and choosing the matches based on the highest scores. The results of their research showed that it is possible to re-identify individuals from large, sparse datasets with high accuracy, even with a small set of auxiliary data points. Furthermore, they demonstrated that the addition of noise or perturbations to the data, which is often used as a privacy protection measure, was not effective in preventing re-identification attacks.[4]

Ohm's 2009 paper "Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization" built upon Narayanan and Shmatikov's research by further highlighting the limitations of anonymization techniques in protecting privacy in microdata.[5] Ohm argued that current legal and technical approaches to privacy protection were inadequate in the face of deanonymization attacks and called for a more nuanced and comprehensive approach to privacy protection. Like Narayanan and Shmatikov, Ohm used the Netflix movie rating dataset to demonstrate the vulnerabilities of anonymization techniques to deanonymization attacks. Ohm showed that even with very little auxiliary data, individuals in the Netflix dataset could be re-identified with high accuracy, despite the removal of personally identifying information. Ohm argued that current legal frameworks, such as HIPAA and FERPA, which rely on de-identification techniques, were insufficient in protecting privacy in microdata. Ohm also emphasized the importance of transparency and informed consent in privacy protection, calling for greater transparency around the data collection and sharing practices of companies and organizations, and for individuals to have greater control over their personal data.

## 3 DATASETS

The primary dataset we are using is a compilation of movie reviews from the top 4,000 Letterboxd users from Kaggle,[6] which ultimately ended up being around 7,500 users. It is split amongst three files. "ratings_export.csv" lists the users, the movies they reviewed, and the individual reviews. "movie_data.csv" gives more information about each movie in the dataset, such as genre(s), language, description, and a link to their IMDB page, if available. The final file, "users_export.csv," gives more information about the users in the dataset, like their display name and number of reviews. The secondary dataset is a much larger dataset of 162,000 MovieLens users and their reviews of different movies.[7] It is split into different files similarly to the primary dataset. Our goal is to use these two datasets much like Narayanan and Shmatikov used the Netflix Prize dataset and the IMDB dataset. That is, we will anonymize the Letterboxd dataset, and then deanonymize it with the help of the MovieLens dataset. By anonymizing and subsequently deanonymizing the Letterboxd dataset, we hope to learn more about the effectiveness of anonymization and its impacts on a user's data privacy. Additionally, by demonstrating the deanonymization process, we hope to find better ways to anonymize data in order to maintain data utility while increasing the users' privacy.

## 4 METHODOLOGY

Our first goal was to anonymize the Letterboxd dataset. The way we approached this was a standard brute force approach to anonymization techniques, such as Data Generalization, in which we changed one or two columns to more general categories. We used Numpy to allow for conditional indexing and the ability to edit all columns at once very efficiently. This allowed us to edit millions of entries in mere seconds, which was useful for other techniques like Synthetic Data and Data Swapping. After each implementation, we would test the $k$-anonymity and modify the method with varying parameters, such as thresholds, in order to meet our standards.

Next, we looked to deanonymize the newly-anonymized Letterboxd dataset. We did this by utilizing an adapted version of the Scoreboard_RH implementation shown in Figure 1. In summary, the algorithm calculates a score for each pair of users across the main and the auxiliary dataset based on the each user's similarities. It calculates this similarity score by factoring into account various attributes that each data set shares, such as the movies reviewed by the pair, and the similarity of ratings given, and the time windows in which the reviews were made. These scores are then calculated for every single pair in each dataset, and then compared at the end

[3] A. Narayanan and V. Shmatikov, "Robust De-anonymization of Large Sparse Datasets," 2008 IEEE Symposium on Security and Privacy (sp 2008), Oakland, CA, USA, 2008, pp. 111-125, doi: 10.1109/SP.2008.33.

[4] Ibid

[5] Ohm, P. (2010). Broken promises of privacy: Responding to the surprising failure of anonymization. UCLA Law Review, 57(6), 1701-1777.

[6] Learner, Sam. "LETTERBOXD Movie Ratings Data." Kaggle, 22 Mar. 2022, www.kaggle.com/datasets/samlearner/letterboxd-movie-ratings-data.

[7] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872.

of the algorithm. If there is a score that is 1.5 standard deviations over the next best score, it would be considered a match.

We instantiate the scoring function as follows:

$$\text{Score}(\text{aux}, r') = \sum_{i \in \text{supp}(\text{aux})} \text{wt}(i)\left(e^{\frac{\rho_i - \rho'_i}{\sigma_0}} + e^{\frac{d_i - d'_i}{d_0}}\right)$$

where $\text{wt}(i) = \frac{1}{\log|\text{supp}(i)|}$ ($|\text{supp}(i)|$ is the number of subscribers who have rated movie $i$), $\rho_i$ and $d_i$ are the rating and date, respectively, of movie $i$ in the auxiliary information, and $\rho'_i$ and $d'_i$ are the rating and date in the candidate record $r'$.[5] As explained in section 4,

**Figure 1: Scoreboard RH algorithm**

In order to implement this, we chose users from the main Letterboxd dataset and the MovieLens dataset and calculated a score based on a) if they have matching movies, and b) a formulation based on the difference of scores. Unfortunately, the Letterboxd dataset does not include timestamps, forcing us to scrap that part of the algorithm. As mentioned in challenges, pairing each user to millions of ratings proved to be challenging in terms of time complexity, so we decided to limit the traversable space by choosing only rare movies, or movies that are not commonly reviewed within the auxiliary dataset. This allowed us to iterate a much more manageable space, yet it was still difficult to go through all users in the main dataset. However, with this implementation, we were able to match some users to those in the auxiliary dataset.

# 5 RESULTS

## 5.1 Anonymization of Letterboxd Dataset

Our first goal was to anonymize the Letterboxd dataset so that we could deanonymize it later and make generalizations about the dataset using an auxiliary dataset to match individuals. In order to do this, we implemented the following anonymization techniques:

- Data Generalization
- Data Swapping
- Synthetic Data

In order to check how anonymized our techniques were, we implemented a $k$-anonymity function for our dataset. This function iterates through the dataset and returns the minimum rows where quasi-identifiers are the same. Within this case, we treated the username as the sensitive data and movie rows and the others columns (the movie identifier and rating) as the quasi-identifiers. Since our goal was also to compare various anonymization techniques and how they fare, we also implemented an average k-anonymity function to see the performance of these anonymization methods if the minimums are low. Here are the results for average k-anonymity for the various anonymization techniques:

| Method | Average K-anon |
|---|---|
| Control | 12.38 |
| Data Generalization | 527.65 |
| Data Swapping (1 Million Swaps) | 11.91 |
| Synthetic Data(10 Million Additions) | 6.92 |
| Improved Data Generalization | 24030.73 |

We elected not to show the standard $k$-anonymity for each anonymization technique, as for each one by themselves, they all had at least one unique combination. However, there are some interesting things to note, specifically that Data Swapping did not increase $k$-anonymity, but might still be useful in other ways, such as protecting users data in case of leaks, etc. Additionally, Synthetic Data did not increase average $k$-anonymity for each combination but actually decreased it, which might suggest that the movie space is so large that even ten million new entries would only add new unique combinations of movie and ratings. We found that Data Generalization performed the best in maximizing $k$-anonymity. Within our dataset, we replaced movie ID's with movie genres and replaced ratings that ranged from 1-10 to three buckets of ratings (1-3, 4-7, 8-10). For our initial implementation, Figure 2 shows the most popular genres and ratings for them.

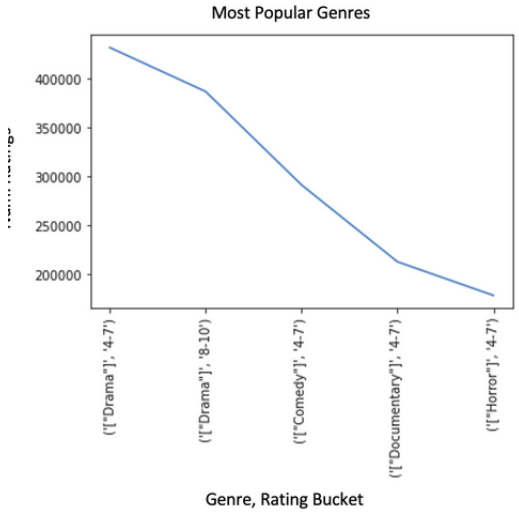


**Figure 2: Most Popular Genres**

Unsurprisingly, single-genre movies were the most popular, and ratings that were average were also the most common. We can draw initial conclusions that users potentially rate movies on some sort of distribution that centers around an average score.

The figure above shows the minimum $k$-anonymity as one for select genres. These are only a few combinations that resulted in a $k$-anonymity of 1. In fact, there were 3,459 combinations of quasi-identifiers that were unique (out of around 21,000 combinations). In other words, around 16% of the dataset is unique entries. In the challenges section, we discuss why this was the case and how we revised the Data Generalization method to increase $k$-anonymity. For our improved Data Generalization method, we found that $k$-anonymity for every possible combination of quasi-identifiers increased, particularly for those that were on the lower end. With this change, there were only ten unique combinations of genres and ratings, and even that could be explained by initializing weights and needing iterations to converge. These are the updated most popular genres:

This figure has two unique combinations where the genre is empty. There are two possible reasons for this. The first potential reason is that the movie lacked a genre in the dataset. The second is that the genre was niche and could not cross the threshold. Furthermore, we conducted experiments to compare possible thresholds

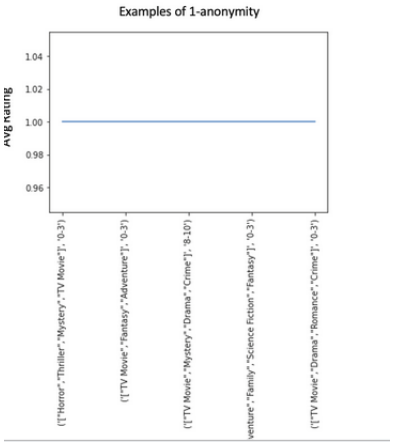


**Figure 3: Examples of One K-Anonymity Genres**

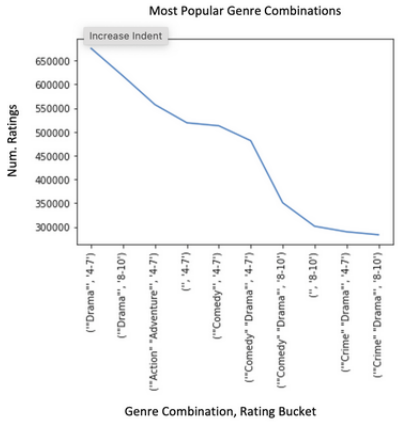


**Figure 4: New Most Popular Genres**

that would increase $k$-anonymity. The following are our results in order to compare data utility and $k$-anonymity.

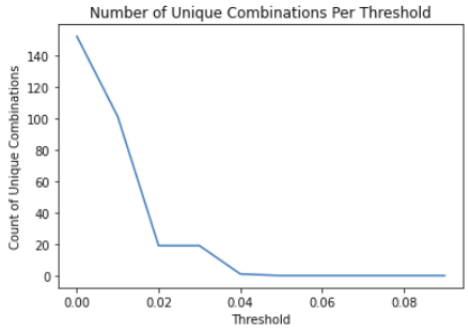


**Figure 5: Unique Combinations Across Thresholds**

The graph above represents the number of unique combinations which could be seen as converging to 0 at higher thresholds. The reasoning behind this is that only popular genres will exist at higher thresholds, which would force a higher $k$-anonymity. The corresponding number of combinations that exist (shown in Figure 6) will represent why data utility is lost because of this increase.

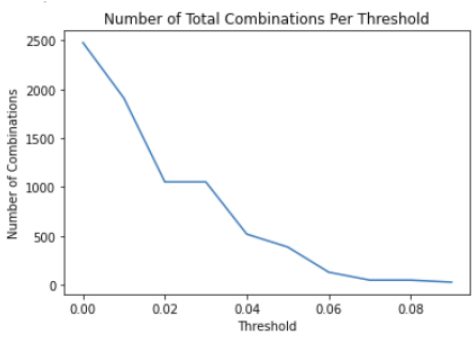


**Figure 6: Total Combinations Across Thresholds**

For the graph above, the total number of combinations for quasi-identifiers steeply drops off and looks to be converging to solidarity. This represents the trade off of data utility as $k$-anonymity increases, though if all movies are classified by just one genre, the usefulness and meaning of the dataset gets lost. Additionally, Figure 7 shows the "average" $k$-anonymity to view general trends of how well the anonymization method is performing.

It is interesting to note that all of these functions are monotonic, and for the unique and total combinations, they are purely decreasing. In fact, some of these graphs might suggest exponential growth/decay, which could be further proven through regression. This suggests that it could be possible to calculate a monotonic increasing function for data utility and then to calculate the ideal balance between data utility and threshold to maximize utility and $k$-anonymity with a simple derivative.

## 5.2 Deanonymization of Letterboxd Dataset

In terms of deanonymization, we quickly realized that implementing the full Scoreboard_RH function would be way too computationally expensive to fully use. With our main dataset containing 7,500

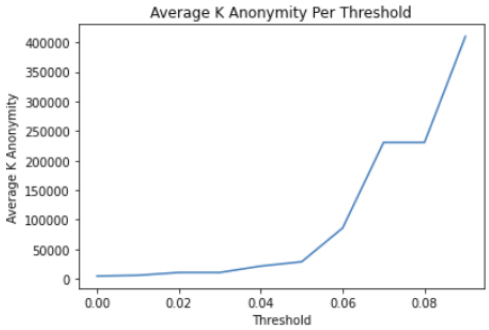


**Figure 7: Average K Anonymity Across Thresholds**

users and our auxiliary dataset containing 162,500 users, we approximated that the full Scoreboard_RH function would take around 220 years to complete. We then decided to cut down the number of users from the main dataset and try to deanonymize 100 users. This decreased the runtime to about 17 years; still way too long. We then toyed with the idea of reducing the number of users in the auxiliary dataset, but decided not to, as this might eliminate potential matches.

We then took a different approach: in addition to filtering users from the main dataset, we also filtered movies as well, only looking at the 10,000 least commonly reviewed movies in the dataset, as rare movies are more likely to narrow our user-matching search down. However, of the 10,000 least common movies in the Letterboxd dataset, only about 600 showed up in the MovieLens dataset. This did not result in any matches that the Scoreboard_RH algorithm was confident in. Then, we tried to use the 100 most popular movies in the Letterboxd dataset, as all of them appeared in the MovieLens dataset. However, this did not narrow our number of users at all, and in fact complicated the search. Finally, we decided to use the 10,000 least common movies that appeared in both datasets. This got some potential scores between users, but none of the scores were good enough to make the algorithm confident in picking them.

Finally, we made the decision to change Narayanan and Shmatikov's Scoreboard_RH function. In their original function, a score must be 1.5 times the standard deviation greater than the next highest score in order to be declared a match. However, due to how sparse these datasets are, this was not resulting in any matches between Letterboxd and MovieLens. Instead, we just declared the highest score between users a match. Then, when reduced to five users from Letterboxd, we were able to match them with users in the MovieLens dataset:

| Letterboxd User | MovieLens User |
|---|---|
| deathproof | 138990 |
| superpulse | 72315 |
| lilfilm | 14610 |
| ianamurray | 83426 |
| punchdrunklizzy | 60656 |

This is our proof of concept that our modified Scoreboard_RH function is able to pair users across datasets, although it is very difficult to fully implement. Additionally, in order to get these results, we had to reduce our anonymization methods to only User Masking and Rating Generalization. If the Scoreboard_RH function were to try to deanonymize a well-anonymized, expansively large dataset using another large database, it would simply be too computationally expensive to be worth it.

## 6 LIMITATIONS

It is important to highlight certain limitations of our methodology. When we initially tried to deanonymize our dataset, we used a combination of the anonymization techniques described above, from data swapping to generalization to creating synthetic data. However, when we implemented a rigorous combination of these, we were not able to find any potential matches. This might also

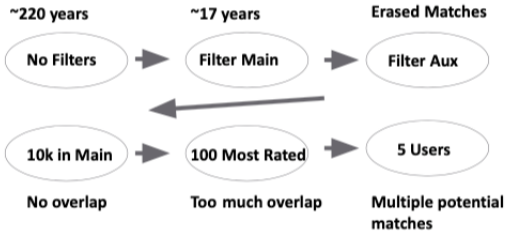


**Figure 8: Progression to Reduce Runtime**

be due to the anonymization of the auxiliary dataset, which could have impeded matching individuals. Thus, in order to prove that the deanonymization algorithm worked as a proof of concept, we scaled back the deanonymization, such that we only implemented the User Masking and Data Generalization methods. As mentioned in previous sections, Data Generalization proved to be the most effective anonymization algorithm in terms of average $k$-anonymity, so there was still a significant amount of anonymization done to the dataset before the deanonymization process started.

It is also important to mention that our results from the deanonymization methods contain some limitations as well. Initially, we planned to deanonymize the full dataset to create as many potential matches as possible, however, due to hardware limitations and algorithmic inefficiencies, we were only able to run the algorithm on a small subset of both the main and auxiliary dataset to create these potential matches. Due to the size of both the main and auxiliary sets, along with the need to compare one user in main to every user in the auxiliary, we estimated that a full search would take approximately 220 years to complete. Furthermore, the way in which the Scoreboard_RH function initially works is that it only considers two users a match across datasets if their matching score was greater than 1.5 standard deviations of the next best matching score. When fit with this criteria on our smaller subset of data, we were not able to create any matches. Thus, we altered the algorithm such that if there was a score at all (which was rare), it would be considered a match. Although this seems like a huge blocker in terms of claiming viable results, the method in which we narrowed down the datasets and filtered through the matches made it incredibly difficult for any two users to be considered a match at all. Two users would have to have watched and rated the same exact set of movies, and rated all of the movies within the same range of rating. This effectively eliminated most matches for us, and allowed us to claim 5 potential matches with our deanonymization algorithm.

## 7 DISCUSSION

### 7.1 Need for Improved Anonymization Techniques

Of the few techniques that we tried and tested, Data Generalization proved to be the most effective anonymization technique, with an average $k$-anonymity score of 527.65 (compared to the next best technique, Data Swapping, that had an average $k$-anonymity score of 11.91). Despite the success of the Data Generalization technique, we found that the utility was being compromised with better

anonymity due to the replacement of movie names for genres. Future work would lead us to try to find a more effective metric to measure the effectiveness of various combinations of anonymization techniques, and to explore the effectiveness of different techniques as well, such as tokenization, aggregation, or differential privacy while maintaining utility.

### 7.2 Privacy Risks

Although we were only able to show proof of concept for the deanonymization algorithm, it demonstrates that re-identification attacks could be carried out with a high degree of accuracy, even on supposedly anonymized data. Our project highlights the privacy risks associated with data sharing and the need for effective anonymization techniques before data. Future research would first require us to attempt to use the Scoreboard_RH and anonymize a larger subset of users with the same level of accuracy as Narayanan's work. In order to do this, we would use the MovieLens dataset as the primary dataset and the IMDB dataset as the auxiliary dataset. This would represent a significant step forward for two reasons. First, the IMDB dataset has already been deanonymized because the accounts are public, providing an excellent opportunity to use it as a supplementary dataset for finding matches. Second, both the IMDB and MovieLens datasets contain timestamps for all reviews, which is an additional attribute that could have been utilized to create matches across the datasets. This would have increased our confidence in the matches since we would have had an additional attribute to filter matches on. Then, once this is accomplished on these two datasets, we would aim to use the same, or lightly adapted version of the scoreboard_RH function on various realms of sparse public data that aren't movie related. Due to this possible deanonymization, we recommend utilizing steps from "De-identification methods for open health data: The case of the Heritage Health Prize claims dataset" such as removing direct identifiers and generalization of indirect identifiers.[8] For example, within our main dataset, we only had access to the user's username. Although there is no direct connection to user's real names, usernames often can be traced back to personal identity. Possibly generalizing the usernames to numbers such as that in the MovieLens dataset could be a good method to eliminate deanonymization.

### 7.3 Legal and Ethical Implications

The ability to re-identify individuals from supposedly anonymized data raises legal and ethical questions about data privacy, ownership, and control. For example, if a company shares anonymized data that can be re-identified, is it still considered anonymous? Should individuals have the right to control their own data and decide how it is used?

## 8 CHALLENGES

The most important limitation we faced was our hardware, considering the processing time to run the Scoreboard_RH algorithm was not realistic or feasible to retrieve the results we expected. Despite reiterating our code to create the most efficient subroutines, the sheer amount of pairings dominated our runtime. Furthermore, considering the size of the datasets and the amount of power this would take, we decided not to run the code locally, instead opting to use Google Colab as a medium in which we could all work on the report together and to take advantage of Google's free allocation of RAM. However even with Google Colab's 12 GB of RAM, an amount likely greater than most available to local computers, we were unable to process all the data. To overcome these issues, we narrowed down the dataset to only five users in the main dataset, and ran the algorithm on the reviews in the auxiliary dataset that had the same common movies such that we could potentially find some matches. Although it may have been ambitious to try and deanonymize the entire dataset, we were definitely hoping to get more definitive results regarding the matches.

Another limitation we faced was the actual usability of the datasets when it came to deanonymizing the datasets. Considering MovieLens and Letterboxd use different rating scales, only MovieLens contained timestamps, and they were both incredibly large, it was very difficult to run even the smallest of mutations or exploration on the data, not to mention comparing the users and actually running the algorithm.

## 9 CONCLUSION

Through our rigorous work with the MovieLens and Letterboxd datasets, we were able to arrive at the following conclusions. In terms of anonymization, a combination of techniques is necessary to enforce better anonymization of the datasets, whether it be through data swapping, generalization, creating synthetic data, or another technique not mentioned in this paper. With all (or even some) of these techniques implemented, it makes deanonymization very difficult, but it also causes utility to decrease. With something as simple as generalizing the names of movies, the algorithm used by Narayanan and Shmatikov would not be as useful because titles could not be matched. In terms of the deanonymization process, using Narayanan and Shmatikov's Scoreboard_RH algorithm, we were able to create potential matches as a proof of concept for the effectiveness of this algorithm and to prove its robustness to apply to different datasets. Considering our results, we conclude that in order to protect individuals' privacy in public datasets, the stakeholders that release these datasets must go beyond traditional anonymization methods and consider other strategies before releasing them to the public.

## REFERENCES


[EDN12] Khaled El Emam, Fida K Dankar, and Adam Neisa. "De-identification methods for open health data: The case of the Heritage Health Prize claims dataset". In: *Journal of medical Internet research* 14.1 (2012), e33. DOI: 10.2196/jmir.1863.

[HK15] F. Maxwell Harper and Joseph A. Konstan. "The MovieLens Datasets: History and Context". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5.4 (2015), 19:1–19:19. DOI: 10.1145/2827872.

[IMD] IMDb. *IMDb Dataset Details*. https://www.imdb.com/interfaces/. Accessed: March 16, 2023.

[NS08] Arvind Narayanan and Vitaly Shmatikov. "Robust de-anonymization of large sparse datasets". In: *2008 IEEE Symposium on Security and Privacy (Sp 2008)*. 2008. DOI: 10.1109/sp.2008.33.

[Ohm10] Paul Ohm. "Broken promises of privacy: Responding to the surprising failure of anonymization". In: *UCLA Law Review* 57.6 (2010), pp. 1701–1.


[8] El Emam, K., Dankar, F. K., Neisa, A. (2012). De-identification methods for open health data: The case of the Heritage Health Prize claims dataset. Journal of medical Internet research, 14(1), e33. https://doi.org/10.2196/jmir.1863

[Swe02] Latanya Sweeney. “k-anonymity: A model for protecting privacy”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.5 (2002), pp. 557–570.