Mitigating Regional Bias in Music Recommendation Systems

Joshua Brusewitz jbrusewi@ucsd.edu
Samantha Horio shorio@ucsd.edu
Shivani Suthar ssuthar@ucsd.edu
Natalie Wu n3wu@ucsd.edu

Mentor: Emily Ramond
eramond@deloitte.com

Abstract

In pursuit of popularity and chart success, global music artists frequently adapt their music to conform to Western industry standards, which possess the most cultural and economic influence. As the industry shifts from traditional physical sales to digital streaming services, algorithmic bias in music recommendation systems has become increasingly prevalent. This bias occurs when the algorithms tasked to recommend songs to users inadvertently favor specific genres or styles of music. Given the nascent state of research on regional bias, our objective was to build a recommendation system designed to preserve cultural uniqueness and diversity in the global music industry by recommending artists from underrepresented countries while accommodating to user taste via bias mitigation.

Website: https://ssutharucsd.github.io/
Code: https://github.com/jebrus/DSC-180B-Final-Project

Contents

1 Introduction

The evolution of popular music charts (e.g. Billboard Top 100) from a simple ranking system to an encapsulation of complex indicators (i.e. market trends, consumption patterns, cultural preferences) has significantly impacted the music industry. These charts not only actively define what is considered popular or mainstream, but also exert pressure on global artists to conform to the preferences and trends that dominate the popular Western charts to an extent in order to achieve commercial success. Consequently, this adaptation leads to a homogenization of music, which erodes cultural uniqueness and diversity within the global music industry. Furthermore, with the industry's shift from traditional physical sales to digital streaming services, the algorithms used in music recommendations now introduce bias and discrimination.

1.1 Background

Digital streaming services have revolutionized music listening habits, enabling users to easily discover new artists and tracks. These services make use of recommendation systems, which make use of user preferences and listening history to suggest music. By employing collaborative filtering, which recommends music based on similar user tastes, and content-based filtering, which focuses on the music's intrinsic properties, they strive to enhance the user experience. However, these algorithms often suffer from algorithmic bias, which can limit exposure to a diverse range of music by favoring certain artists and genres, particularly those from Western countries. Such biases exacerbate the overarching problem by promoting a narrow range of music styles and origins, further impacting cultural representation in music listening habits. As digital streaming services become the norm, these algorithms start to dictate users' listening habits, underscoring the need to explore bias mitigation in recommendation systems to promote inclusivity in the music industry. In response, our team aimed to develop a music recommendation system that mitigates regional bias by recommending artists based on their country of origin to preserve diversity in the global music industry. As a result, we were able to bring more awareness and attention to artists from underrepresented countries, where our product successfully promoted cultural preservation and equitable representation in the music industry, specifically in the digital music landscape.

1.2 Literature Review

To build our music recommendation system, we drew from previous literature in the field of recommendation systems to inform our methodology– namely, the baseline model development and bias mitigation process.

For baseline model development, it was found Collaborative filtering (CF) methods are the most widely used in current literature. CF methods provide recommendations using collective preferences of users within the same group. For example, if there are two users

in the same "preference group", and one of the users likes a particular item, then it is likely the other user will like it as well. This is the core ideology of CF– users/people placed in the same group will like the same things.

The paper "Musical Instruments Recommendation System Using Collaborative Filtering and KNN" (Puspita et al., n.d.), uses this collaborative filtering ideology along with a k-nearest-neighbors (KNN) algorithm to build a musical instrument recommendation system. In short, a KNN algorithm places similar users in the same preference group together. The dataset used in this paper was a musical instrument review dataset from Kaggle (i.e. a data science website that provides datasets on a multitude of topics), which included a "reviewID" column that contained a review ID for each user that left a review, an "asin" column that had information about the product ID of each musical instrument, and an "overall" column that contained a rating of the musical instrument given by the reviewer. Using the "asin" and "overall" columns, users were placed into preference groups and their similarities were calculated using three classic similarity metrics– cosine similarity[1], mean squared difference (MSD)[2], and Pearson correlation coefficient[3]. A different KNN model was built using each of these similarity metrics and these three models were compared using classic performance metrics– mean absolute error (MAE)[4], and root mean squared error (RMSE)[5]. They found that using the Pearson correlation coefficient to determine the similarity for the preference groups performed the best.

Another paper that utilized collaborative filtering was "Movies recommendation system using collaborative filtering and k-means" (Phorasim et al., 2017), where a movie recommendation system was built. They utilized a k-means-clustering algorithm which groups users into clusters where the users within each cluster should be as similar as possible. The dataset used in this paper consisted of movie ratings data from the MovieLens project, which included a column for user IDs, movie IDs, and their corresponding ratings. To develop the k-means-clustering algorithm, users were categorized into 10 clusters based on their movie ratings, using Euclidean distance[6] to measure the similarity between them. After the users were placed into clusters, the Pearson correlation coefficient was used to further determine the most similar users within the clusters so that the most accurate recommendations could be provided using the CF assumption that similar users will have similar preferences. They

[1]Cosine Similarity: A metric indicating the similarity between two users' interests based on the items they've rated by treating their ratings as angular directions, with values closer to 1 suggests higher similarity.

[2]Mean Squared Difference (MSD): A measure of how consistently similar or different the users' ratings are, where lower values mean more similarity; calculated by averaging the square rating differences between two users.

[3]Pearson Correlation Coefficient: Indicates the degree to which two users' ratings are linearly related, with values ranging from -1 (negative correlation, i.e. points move in opposite directions) to 1 (positive correlation, i.e. points move together). For recommendation systems, it helps us see if two users tend to rate items similarly.

[4]Mean Absolute Error (MAE): Represents the average error in predictions (difference between the predicted ratings and the actual ratings given by users), with lower values indicating more accurate predictions.

[5]Root Mean Squared Error (RMSE): Similar to MAE but gives more weight to larger errors, with lower values indicating more accurate predictions. Calculated by squaring the differences between predicted and actual ratings, get the average, and then take its square root.

[6]Euclidean Distance: Measures the straight-line distance between two points with shorter distances indicating more similar tastes between the users.

found that using k-means clustering significantly improved the recommendation accuracy and efficiency, compared to previous papers mentioned in this study that used a classic CF method based on directly comparing user ratings of items without the preliminary step of clustering users into groups.

Lastly, a third paper that leveraged the collaborative filtering methodology was "Using SVD and demographic data for the enhancement of generalized Collaborative Filtering" (Vozalis, 2007). In this paper, they used a CF technique known as singular value decomposition (SVD) to build a movie recommendation. SVD is a method where large datasets are broken down into simpler tables, known as matrices. They are broken down based on latent (i.e. hidden) features/patterns in the data through a mathematical process known as factorization. The dataset used in this paper consisted of user IDs, movie IDs, and their corresponding ratings. It also incorporated demographic information about the users, such as age, gender, and occupation. Then, the SVD algorithm was used to factorize the user-movie rating data into three matrices, capturing latent features/patterns associated with both users and items. This process is known as dimensionality reduction and helps address some of the inherent challenges in collaborative filtering, such as sparsity which is not having a sufficient amount of data, and scalability which is the ability to manage large amounts of data without suffering performance losses. When evaluated with MAE, the integration of SVD and demographic data significantly improved the accuracy of the recommendation system.

Overall, these three papers helped to inform our baseline model development.

For our bias mitigation process, after much investigation of past literature, we found that currently, the primary areas of bias mitigation in music recommendation systems center around gender bias and popularity bias. However, there are no current studies that focus specifically on mitigating regional bias in music recommendation systems which is where our project helps to fill in this gap. Nonetheless, we still drew insights and inspiration from the existing literature to aid in our bias mitigation process.

The paper "Strategies for Mitigating Artist Gender Bias in Music Recommendation: A Simulation Study" (Bauer et al., n.d.) discussed various methods to mitigate gender bias in music recommendation systems. The dataset they used was a subset of a Last.fm dataset with added information about artists' genders. The first method this paper explored was moving the first recommendation of a female artist that popped up on a user's recommended list to the top of their recommendations. The second method they used is a method known as alternating least squares (ALS) which penalizes gender imbalance by using a matrix factorization process, similar to that used in SVD. This works by adjusting the recommendation algorithm to favor items representing underrepresented genders– in this case, women– thereby increasing their visibility in the recommendation list. The main results from this study were that moving the first recommendation of a woman to the first position positively impacted the recommendation of female artists with the precision of the algorithm being only minimally impacted. It was also found that penalizing the imbalances helped in recommending more diverse groups of female artists. However, this took a hit on the algorithm's accuracy– a common issue which seems to present in current music recommendation systems of balancing algorithmic fairness and accuracy.

The paper "Mitigating Popularity Bias in Music Recommendation Systems: Effects on Fair

Exposure, User Perception, and Motivation for Exploration" (Ungruh, 2023), focused on addressing the issue of popularity bias in music recommendation systems. In order to do so, they used a novel recommendation algorithm, RankALS, along with two established popularity bias mitigation techniques– one aimed at promoting fairness towards artists (i.e. FAIR) and another focused on user fairness (i.e. calibrated popularity). RankALS essentially adjusts the ranking of recommended items by penalizing the over-representation of popular items. The FAIR method aims to achieve artist fairness by ensuring that songs from less popular artists have a fair chance of being recommended by increasing their weight in the algorithm. The calibrated popularity technique focuses on user fairness by calibrating recommendations to include items that match the users' historical preference patterns in terms of popularity. The main results from this study were that the FAIR method was particularly successful in increasing the visibility and potential discovery of less popular artists. However, it came at the cost of algorithmic accuracy, similar to the previous paper. The calibrated popularity method successfully aligned recommendations with users' historical listening patterns but still seemed to be biased in favor of popular artists.

After reading these papers, we determined that using reranking– similar to the idea of reweighting from the popularity bias mitigation paper– , along with the implementation of other bias mitigation techniques, would be the best for our study to not only mitigate regional bias, but also take steps towards closing the gap between fairness and accuracy– a current issue in the field of recommendation systems.

1.3 Data Description

Our music recommendation system leverages two datasets categorized into a user dataset and an artist dataset, derived from Last.fm and MusicBrainz. Last.fm is a music service that uses a technique called "scrobbling" to track users' music listening habits across various platforms and build a detailed profile of their musical tastes. Such is used to recommend new music, connect users with similar interests, and provide personalized music charts. On the other hand, MusicBrainz is an open, comprehensive database that collects music metadata from the community and makes it available to the public for various purposes. The integration of the user and artist datasets, procured from the most comprehensive and readily accessible online source of recent user listening data, into a music recommendation system allows for the mitigation of regional by utilizing user preferences and artists' nationalities.

With the Last.fm API, we acquired a user dataset by first selecting a starting user and adding a randomized list of that user's friends to the total user list. Consequently, we added this randomized list to the front of the queue and grabbed the next user's friends– these steps were repeated until there was a sufficient number of users, which was roughly 900,000. Additionally, we pulled each user's top artists by looping through the user list that was just obtained and skipped users with no top artists, which we then reformatted into a data frame to work with.

This provides four attributes: `user`, representing the user's ID; `artist_name`, the name of the artist; `play_count`, indicating the number of times the user has played the artist's

tracks; and `artist_url`, the URL to the artist's Last.fm page. Together, these attributes detail the user's top artists, their play counts, and direct links to the artists' Last.fm profiles.

Similarly, we got an artist dataset by using both the Last.fm API and MusicBrainz API. Using the `artist_name` attribute from the previous dataset, we checked for each unique artist on Last.fm while accounting for potential formatting differences; if the artist was there, we pulled their MusicBrainz ID and queried for the artist's information, but if they either had no listed ID on Last.fm or is not there at all, their name was logged. We then looped over the list of logged names to pull them from MusicBrainz using their API's search functionality while checking if the logged name was exactly the same as the name in the search result. There were a few artists we were not able to pull because of formatting issues, but we determined that the number is negligible in the overall picture.

This provides two attributes: `artist_name`, the name of the artist, and `country_code`, the code of the country of their nationality. For readability purposes, we converted the country codes to the name of the actual country.

2 Methods

2.1 Exploratory Data Analysis Insights

Through our initial EDA of the dataset, we found that the vast majority of users have a full top 50 artists, meaning users with lower amounts will not affect predictions as much because such numbers can be attributed to their low usage of last.fm rather than the total sum of their listening habits. Additionally, it seems that most users do not have many plays on the artists— around the thousands or tens of thousands— and only a few superusers have an extensive amount of plays, which is something to take into account in our recommendation system.

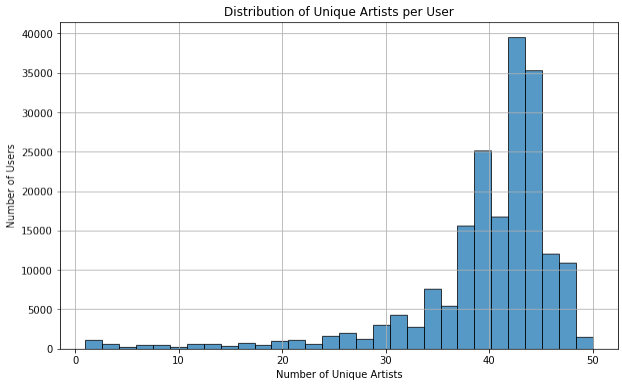

Figure 1: This plot shows the distribution of Unique Artists per User. Most users appear to listen to between 10 and 30 unique artists, indicating how the bulk of our user base does with artist diversity in their listening habits. A few explore over 40 unique artists, suggesting a broader taste or greater openness to exploration.

Taking these low-usage players into account, we filtered our user dataset so that all users with less than 50,000 total plays on their favorite artists within the dataset, as this eliminated nearly all low favorite-count users without eliminating too many users and left us with 6,904 total users. Artists with low favorite counts would create a biased model if they were included. For example, if a player only listens to two artists– which most likely indicates low usage of Last.fm–, the model would produce a correlation between those artists that would not reflect the general population's opinion and create strong bias in the recommendation system since we do not weigh users by play count. Furthermore, we filtered out all artists who did not have a location on file from the artist dataset, as they would obstruct the bias mitigation process.

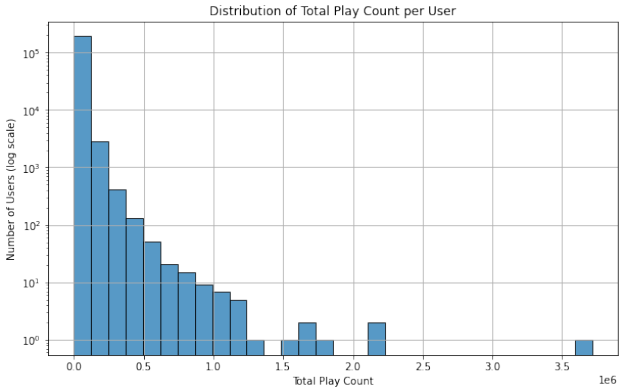

Figure 2: This plot shows the Distribution of Play Count per User. User engagement varies widely; high play counts may reflect fixed preferences, making it hard to make diverse recommendations, whereas users with fewer plays could be more receptive to exploring new, including underrepresented, artists.

Moreover, our analysis revealed a significant concentration of popular artists originating from the U.S., U.K., and Canada, corroborating our hypothesis of inherent regional bias in the existing dataset. There is a disproportionate representation of these countries in terms of artist popularity, as evidenced by the skewed distribution of play counts towards artists from these countries. Such trends introduce limitations of the diversity in music recommendations to users.

Based on the listener count as well as expected end-user familiarity, we decided to consider the overrepresented countries to be the U.S., U.K., and Canada since they made up an overwhelmingly large portion of the data, and the rest of the countries made up the underrepresented group. In light of these findings, we began to work towards the development of a model specifically designed to mitigate regional bias, keeping those overrepresented countries in mind. Guided by our insights, we set a clear objective to ensure that artists from underrepresented countries gain visibility.

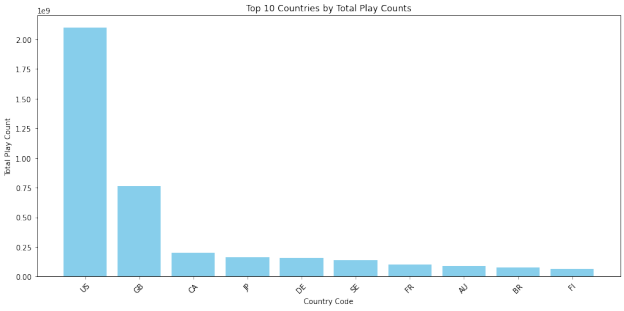

Figure 3: This plot shows the top 10 countries by total play counts. The U.S. disproportionately dominates the music industry having a much larger total play count compared to the others. GB and CA, respectively the U.K. and Canada, follow as the most popular countries.

2.2 Baseline Model Development

From evaluating the aforementioned studies, we determined that using a collaborative filtering approach was an appropriate approach we could implement for our baseline model since it is the most widely used and established method in the field of recommendation systems.

In addition, we determined that utilizing an SVD model would work best with our dataset. As mentioned in the literature review, an SVD model works by breaking down larger datasets into simpler tables, known as matrices, based on latent features/patterns it discovers through a mathematical process known as factorization. This entire process is known as dimensionality reduction.

One of the reasons we decided to leverage SVD is that while KNN and k-means-clusters were viable, they did not provide the benefit of computational efficiency that comes from the dimensionality reduction feature of SVD. In addition, SVD's ability to discover latent features was particularly important to us as it provides insight into more nuanced relationships between users and artists, improving accuracy.

In order to develop our baseline SVD model, we applied the `TruncatedSVD` method from Python's `sklearn.decomposition` module to decompose our user-artist interaction matrix into lower-dimensional matrices representing latent factors associated with users and artists. This allowed us to predict how much a user with an inputted set of liked artists will enjoy every other artist in our dataset. Specifically, we chose 50 as the number of latent factors for our model, since that was the "sweet spot" between balancing computational efficiency and quality of predictions, and as mentioned, we were also trying to take steps to bridge the gap between fairness and accuracy in our project.

Lastly, it is important to note that an assumption we made in implementing our SVD model is that which is used in collaborative filtering, which is that we assumed that users with similar music listening patterns are likely to have similar preferences.

2.3 Bias Mitigation

We chose to de-bias our SVD model using the post-processing technique known as re-ranking. This method is applied after model training and reweighs recommended items to prioritize underrepresented groups and diversify the output, guided by fairness criteria like demographic parity. It aims to mitigate biases against underrepresented or marginalized groups that may exist in the model's initial output. The re-ranking process involving underrepresented countries as opposed to filtering out artists from overrepresented ones was chosen because we wanted to leave the opportunity for the model to recommend an artist from an overrepresented country, should the weight for that artist be so high it gets in despite the reweighing. This is to ensure that an artist that is 100% perfect for a user is still recommended regardless of whether they are overrepresented– although most artists are not absolutely perfect.

The re-ranking process first involves our SVD model initially assigning every artist a value corresponding to how likely it is that a user will enjoy their music. Then, the value for each artist from an underrepresented country is multiplied by a hyperparameter `underrepresented_weight` with a value greater than or equal to one, which allows us to reweigh our model to prioritize artists from underrepresented countries. As a result, we get the artist's `reweighted_score`, which is equal to the product of the `unweighed_score` and `underrepresented_weight`. Increases in `underrepresented_weight` led to significant changes in the results, which is why we opted for this linear relationship, as a stronger relationship between the two would likely lead to overweight results.

Additionally, we multiply this `reweighted_score` by `popularity_rank`, a value that lowers the artist's rank based on the proportion of the total favorites from their country (i.e. how many users favorited the artist). In other words, scores for popular artists within a country would be lowered accordingly, e.g. Taylor Swift's score would be much lower than At The Drive-In's within the U.S. as she has far more users marking her as a favorite artist in the dataset. This value was determined by a hyperparameter called `popularity_weight`, which is calculated as so:

$$(1 - (\text{number of users who favorited the artist} - \text{number of favorites among all artists within the country}))^{\text{popularity weight}} \quad (1)$$

Altogether, we get the popularity-weighted score, where there's an exponential relationship between `popularity_rank` and `popularity_weight`. Since most of the proportions were extremely close to one, the reweighing would have contributed very little if we did not amplify the minor differences between them; thus, we used an exponential relationship. This metric was added to ensure that artists of all sizes are recommended.Hence, the two hyperparameters `underrepresented_weight` and `popularity_weight` were used to tune our model.

To evaluate the efficacy and fairness of our SVD model, we employed multiple metrics, namely Disparate Impact, Average Match Score, and Mean Popularity Rank.

Disparate Impact is a widely used fairness metric in industry, which assesses if an algorithm unintentionally discriminates against certain groups more than others and reveals poten-

tial bias in positive outcomes. It does so by dividing the percent of the underprivileged group[7] that receives a positive outcome by the percent of the overprivileged group[8] that receives a positive outcome. In this case, this entails artists from underrepresented and artists from overrepresented countries respectively, where the positive outcome would be the event in which an artist gets recommended. If this value is below one, this means that the overprivileged group has a higher rate of positive outcomes, whereas a value of exactly one means a perfect balance between both groups. While one is considered the ideal value, most conventionally aim for 3/4, or 75% as their ideal value/best case. However, we chose to interpret values that are greater than one to be favorable, as we specifically want the underprivileged group (i.e. artists from underrepresented countries) to be recommended more. This metric was specifically chosen because our topic aligns with the demographic parity framework where we want to systematically support underprivileged groups.

To offer further insights into model performance, we devised the additional metrics– Average Match Score, and Mean Popularity Rank. Average Match Score is the mean of an artist's score that determines how likely a user will like them, and Mean Popularity Rank shows how popular the artists recommended are based on the average number of listeners the recommended artists have within the dataset. The rationale behind Mean Popularity Rank is to ensure that our recommendations extend beyond widely popular artists, as there is no meaning in recommending diverse artists if they are already huge in the U.S. While we did not explicitly set ideal values for these metrics, we preferred a higher Average Match Score, indicating a higher chance of user satisfaction; a lower value for Mean Popularity Rank, signifying more recommendations for underrepresented artists. Hence, from the baseline to the de-biased model, our goal was to observe an increase in Average Match Score and a decline in Mean Popularity Rank, facilitating the overall goal of more equitable music recommendations.

2.4 Web Application and Integration

After successfully mitigating bias in our SVD model, our next step was to transform this advanced model into an accessible tool for the public. The goal was to create a user-friendly web application that allows users to input their favorite artists and receive a list of recommended artists from underrepresented countries, thus promoting global music diversity. This section outlines our approach to developing this web application and integrating our de-biased music recommendation system.

[7]Underprivileged group: a group negatively impacted by algorithmic decisions, experiencing exclusion, bias, or discrimination that results in reduced opportunities, resources, or outcomes compared to others.

[8]Overprivileged group: a group that benefits from algorithmic biases intentionally or unintentionally, leading to a perpetuation of existing inequalities where they gain advantages at the expense of the underprivileged.

2.4.1 Front-End Design and User Interface

To design the web application, we focused on simplicity and usability. Using HTML, CSS (as detailed in the provided `web_app.css`), and JavaScript, we created an intuitive interface that prompts users to enter their favorite artists. The design prioritizes ease of navigation and clarity, ensuring that users of all technical backgrounds can interact with our application without confusion.

2.4.2 Back-End Integration and Flask Framework

For the back-end, we chose Python's Flask framework for its flexibility and ease of integration with our existing Python-based model. The `web_app.py` script acts as the core of our application, handling user requests, processing input through our de-biased SVD model, and returning music recommendations. Flask enabled us to efficiently route user requests and dynamically render the recommendations on the web page.

2.4.3 Model Integration and Response Generation

Integrating the de-biased model into the web application involved serializing the trained SVD model using libraries such as `pickle` or `joblib`. This allowed us to quickly load the model upon application startup and perform real-time predictions based on user input. When a user submits their favorite artists, the application queries the model, which then calculates similarity scores and applies the re-ranking logic to boost artists from underrepresented regions. The result is a curated list of recommended artists, dynamically generated and displayed to the user.

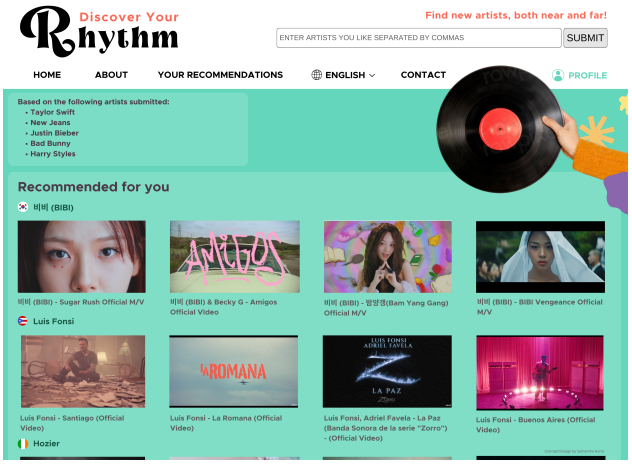

Figure 4: This is proof of the concept of our intended end product, where the user can input their favorite artists and get four music videos from artists our model recommends.

3 Results

After implementing our baseline model, we found that our Disparate Impact value was 0.1890, highlighting a bias against underrepresented artists in comparison to popular artists. Our average match value was 0.0799 indicating that there is an approximately 7.9% chance that a user will like the artist they are recommended. Furthermore, the Mean Popularity Rank value was 426.2001, indicating that artists that are being recommended by our baseline model are rather popular and not likely underrepresented artists.

After hyperparameter tuning, where we set the `underrepresented_weight` to 25 and the `popularity_weight` to 200, we found our Disparate impact improved from 0.1890 to 62.4505, significantly enhancing visibility for underrepresented artists and promoting equitable recommendations. There was a minor decline in the Average Match Score score from 0.0799 to 0.0139, but the difference is likely negligible as it still aligns with our overarching goal of offering a diverse music selection over perfect matches. Additionally, the Mean Popularity Rank significantly improved from 426.2001 to 36.0155, evidencing the model's success in recommending lesser-known artists.

Table 1: Comparison of both models on the specified metrics.

	Baseline Model	Debiased Model
Disparate Impact	0.1890	62.4505
Average Match Score	0.0799	0.0139
Mean Popularity Rank	426.2001	36.0155

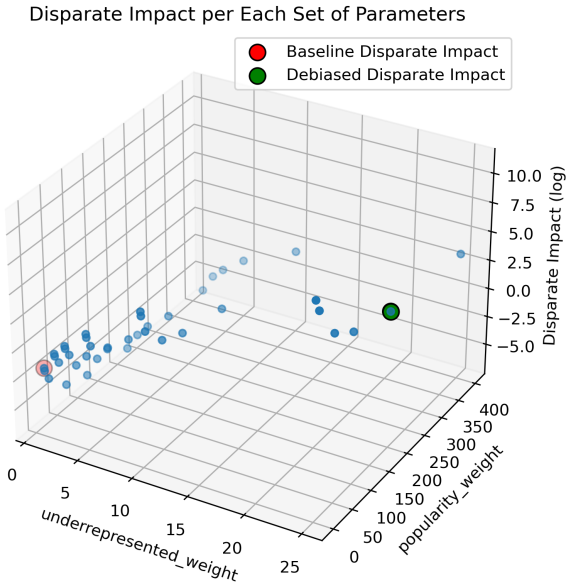

Figure 5: `popularity_weight` and `underrepresented_weight` are plotted against Disparate Impact on a log scale.

Based on Figure 4, it appears that `underrepresented_weight` has a high positive correlation with Disparate Impact, whereas `popularity_weight` has a minimal effect. Notably, after hyperparameter tuning, the outcome for artists from underrepresented countries improved by roughly 300 times in terms of Disparate Impact.

4 Discussion

There was a notable improvement in the Disparate Impact metric, which underscores a shift to a more equitable representation of underrepresented artists and fulfills our objective of building a recommendation system that mitigates regional bias. To reiterate, our objective does not require us to aim for a certain threshold but rather, demonstrate an improvement in Disparate Impact– with higher values indicating better performance– which our results demonstrated. Although the Average Match Score slightly decreased, we chose to prioritize the diversity in artist recommendations over the accuracy of individual matches. On the other hand, Mean Popularity Rank saw a substantial improvement, with the decrease in rank, which corroborates the efficacy of our model in diversifying artist recommendations and increasing exposure of lesser-known artists to users. Hence, we were able to enrich users' music listening experience while mitigating regional bias by successfully recommending artists from underrepresented countries.

Regarding the web application, the development and deployment of our de-biased music recommendation system represent a pivotal step towards addressing the systemic bias prevalent in the music industry, particularly in digital platforms. By intentionally amplifying artists from underrepresented countries, our tool challenges the conventional music discovery paradigms that often favor artists from a handful of Western countries. Our system acts as a bridge connecting users with a diverse range of music traditions, genres, and artists they might not have encountered otherwise. This exposure fosters a greater appreciation and understanding of global cultures, encouraging a more inclusive music ecosystem. Moreover, by prioritizing artists from regions that are typically marginalized in mainstream music recommendation algorithms, we provide these artists with a platform to reach a wider audience. This not only helps in leveling the playing field but also contributes to the artists' potential financial and professional growth.

4.1 Limitations and Next Steps

Given our limited time frame, with a substantial portion dedicated to data collection, we were unable to explore other models and optimize performance as originally intended. Consequently, our efforts went towards the development of a minimum viable product (MVP). These circumstances present multiple areas for improvement and exploration

Moving forward, we would like to explore other methods, to address our lower-than-desired Average Match Score and potentially improve performance in other areas. To achieve this, we plan to integrate hybrid methodologies into our existing SVD model, which includes

adopting strategies from recent research like the approach outlined in the paper "Strategies for Mitigating Artist Gender Bias in Music Recommendation: A Simulation Study" (Bauer et al., n.d.), which involves prioritizing the first recommendation of underrepresented groups by placing it at the forefront– a method shown to impact algorithmic precision minimally. Additionally, we intend to employ the calibrated popularity technique from the popularity technique from the popularity bias mitigation study to accommodate recommendations to be closer to users' tastes.

In addition, we aim to explore other techniques such as neural collaborative filtering (NCF) and content-based filtering. These will allow us to capture more user-song interaction patterns, namely non-linear ones, and provide more curated recommendations based on audio and lyric features, respectively.

Beyond refining our algorithmic approach, we would also like to pursue bias mitigation across several categories in our model. With the multifaceted nature of bias, we would like to extend our efforts to address various prejudices, including but not limited to, gender and genre. These are common areas of bias that come to mind both in general and in the music industry. Such a comprehensive approach is necessary to address underrepresentation.

Along with these technical improvements, we want to refine our web application that employs our model to include additional features to improve user experience and engagement. Altogether, we ultimately wish to foster a more inclusive, diverse, and equitable musical experience, where every artist is given the opportunity to be heard and users can discover music that resonates with them. Enhancing the web application to better utilize user feedback for model retraining and refinement can ensure that the recommendations improve over time. Implementing a more nuanced feedback system could help in capturing user satisfaction more accurately. Developing partnerships with existing music streaming platforms to integrate our de-biased recommendations could also significantly increase the system's impact and reach a broader audience.

4.2 Conclusion

The development and implementation of our debiased model demonstrates significant progress towards our objective of promoting cultural diversity and inclusivity within the music recommendation space. By effectively reducing algorithmic bias, we have broadened discovery channels for users to explore a large array of talented musicians around the globe, thereby enriching their listening experience and enhancing the visibility of underrepresented artists. This project not only showcases the potential for algorithmic solutions to address cultural biases but also lays the groundwork for further research and advancements in equitable recommendation systems. It signifies our objective to build a more inclusive digital music ecosystem, where diversity is recognized.

However, as we progress, it is crucial to acknowledge that "correcting" bias is not a perfect solution. These algorithmic approaches often fail to address the context and nuances of the biases present, which reveals the complexity of absolute fairness which can be a subjective concept depending on the stakeholders and situation. Oftentimes, if not almost

always, there are trade-offs between achieving fairness and optimizing other objectives like accuracy or efficiency, creating this need to strike a balance. Additionally, the pursuit of a "perfectly fair" algorithm is complicated by real-world challenges, including compliance with legal frameworks and ethical guidelines as well as transparency issues. These factors collectively make it difficult to achieve absolute fairness, leading to the development of algorithms that attempt to find a balance among these considerations instead. Therefore, we want to emphasize that building fairer algorithms is an iterative process, not a final solution. Henceforward, we would like to continuously refine our strategies to acknowledge these limitations and navigate through the intricacies of algorithmic fairness and inclusivity, specifically in the demographic parity area to give opportunities to diverse communities and foster equity.

References

1. Bauer, Christine, and Andres Ferraro. "Christine Bauer | EXDIGIT Professor of Interactive Intelligent ..." research gate. Accessed March 9, 2024. https://www.researchgate.net/profile/Christine-Bauer-2.
2. Phorasim, Phongsavanh, and Lasheng Yu. "(PDF) Movies Recommendation System Using Collaborative Filtering ..." Research Gate, 2017. https://www.researchgate.net/publication/314250702_Movies_recommendation_system_using_collaborative_filtering_and_k-means.
3. Puspita, Alfriska, Vynska Permadi, Aliza Anggani, and Edwina Christy. "Musical Instruments Recommendation System Using Collaborative Filtering and KNN." Download.garuda.kemdikbud.go.id. Accessed March 4, 2024. https://download.garuda.kemdikbud.go.id/article.php?article=2495305&val=23800&title=Musical+Instruments+Recommendation+System+Using+Collaborative+Filtering+and+KNN.
4. Ungruh, Robin. "Master Thesis U.S.E." studenttheses.uu.nl, 2023. https://studenttheses.uu.nl/bitstream/handle/20.500.12932/42389/Farahbakhsh,M.F._4306651.pdf?sequence=1&isAllowed=y.
5. Vozalis, M.G., and K.G. Margaritis. "Using SVD and Demographic Data for the Enhancement of Generalized Collaborative Filtering." Information Sciences, March 15, 2007. https://www.sciencedirect.com/science/article/pii/S0020025507001223.

Appendices

This appendix details additional definitions and explanations of any terms or metrics used throughout the report.

1. **Singular Value Decomposition (SVD):** Singular Value Decomposition (SVD) is a mathematical technique used in data science to decompose a matrix into three other matrices, enabling the simplification and enhancement of data processing. In the context of recommendation systems. This dimensionality reduction technique decomposes user-item interaction matrices and reveals essential patterns for predicting user preferences, making it well-suited for high-dimensional data.

$$A = U\Sigma V^* \tag{2}$$

   where A is the original matrix, U is an orthogonal matrix representing the left singular vectors, Σ is a diagonal matrix with singular values, and V^* is the conjugate transpose of V, an orthogonal matrix representing the right singular vectors.

2. **Average Match Score:** The Average Match Score represents the mean score of an artist, quantifying the likelihood that a user will enjoy them. This metric is pivotal in personalizing recommendations to align closely with user preferences.

$$AMS = \frac{1}{N}\sum_{i=1}^{N} S_i \tag{3}$$

   where AMS is the Average Match Score, N is the number of artists, and S_i is the score of the ith artist, indicating the user's likelihood of liking the artist.

3. **Mean Popularity Rank:** The Mean Popularity Rank indicates the relative popularity of recommended artists, calculated based on the average number of listeners for the recommended artists within the dataset. This metric helps gauge the mainstream appeal of the recommendations given to users.

$$MPR = \frac{1}{N}\sum_{i=1}^{N} P_i \tag{4}$$

   where MPR is the Mean Popularity Rank, N is the number of artists recommended, and P_i is the popularity rank of the ith artist, determined by their listener count within the dataset.