Offbeat: A Music Discovery Tool to Diversify Users' Listening on Spotify

Tyler Inness; Waverly Konynenbelt; Drew Mooney; Jeff Rose; Tim Wilcox
{james.inness,waverly,drew.mooney,jrose67,tim.wilcox}@gatech.edu
Georgia Institute of Technology

1 INTRODUCTION

1.1 Motivation/Problem Statement

Music recommendation systems are ubiquitous in today's world of streaming music. Streaming platforms regularly curate playlists for users based on those users' listening behavior. These systems typically aim to provide users with novel song recommendations while still surfacing content those users are likely to listen to. A result of this attempt to balance novelty with familiarity is that music consumers relying on these curated playlists can feel their music all starts to sound the same. Not only does this result in listeners becoming bored, but it effectively limits users from discovering new, enjoyable areas of music that are dissimilar from their current musical interests. Additionally, current music recommendation systems often give little insight into how recommendations are made, which can make it hard for users to know what kind of music they might be missing out on. [16].

1.2 Proposed Solution

To address the above-mentioned shortcomings of traditional recommendation engines, we introduce Offbeat, a tool that enables Spotify users to explore music that is different from what they typically listen to. This approach offers a completely new and innovative platform for discovering music on Spotify by:

- Highlighting clusters of songs that are dissimilar to music a user typically consumes
- Giving the user visibility into how recommendations are made
- Allowing the user to determine how different they want suggested songs to be from what they usually consume

We predict that this new approach will be successful as it will help Spotify users (of which there are more than 200 million, many of whom are looking for new, interesting music) uncover new musical areas of interest and help them understand more deeply the way in which their interests are connected to the recommendations that they see. At the conclusion of our product development, we shared Offbeat with friends, family and colleagues to gain understanding into how successful we were in helping users discover new music.

2 SURVEY

Offbeat combines the areas of *recommendation algorithms*, *clustering*, *visualization* and *interaction*, each of which has a large amount of relevant, published work.

2.1 Recommendation

There is a wide body of research into recommendation systems (both for music and other areas) and how they are currently implemented. Increasingly it is recognized that algorithms set agendas, shape content that is consumed by end users at large and define peoples' tastes [9]. In the world of streaming music this is also true; streaming services and recommendation engines have strong control over what users are exposed to (shaping how much users can actually "discover" on these platforms) [4, 6, 10]. These recommendation services use methods like content-based filtering and collaborative filtering to drive music selection in an attempt to balance users' desires for similarity with the occasional desire to try new things [1, 7]. The main ideas that are implemented in these recommendation systems are tangentially useful to us, but we see a limitation in how traditional recommendation systems surface suggestions to users. With Offbeat, we aim to move in a different direction in multiple ways - both in how we expose some of the inner workings of our algorithm and in how we purposefully surface content that is dissimilar from users' expressed interests.

Some research has been done in an effort to improve recommendation systems by adding new or different features into the underlying algorithm, which could theoretically provide users with better and more relevant recommendations. Some examples of these features are social influence [2], contextual inputs (including extrinsic and intrinsic factors) [16], and song or vocal characteristics [15]. We do not implement external features in our project; instead, we develop our algorithm based on intrinsic song characteristics and assess how each user's personal interests intersect with the larger Spotify ecosystem of songs.

2.2 Clustering

Due to the hierarchical nature of music classification (genres, sub-genres, artists, albums, song), we have built the Offbeat interface in a way that allows for easy traversal and exploration of different hierarchical layers. An algorithm that matches this interaction style very well is hierarchical agglomerative clustering [3, 5]. We implement this modeling technique with our song data, but we do not improve upon the algorithm itself during this project. There are also alternative clustering methods that are possible, such as a divide-and-conquer approach [8] or a vector map which can model similarity by using vector length [12]. However, we do not pursue these approaches, instead we stick with the core agglomerative clustering algorithm in order to realistically finish our model development in the desired timeline.

2.3 Visualization and Interaction

Offbeat uses the data visualization best practice of *data storytelling* to communicate dynamic findings to each user based on their personal data and interaction choices [11]. To allow users to interact with their personal Spotify data and our exploration algorithms, we have built Offbeat as a web-based application that utilizes d3.js [13, 14]. We do not improve upon the d3.js framework; rather, we use its powerful hierarchical layout features to create an interactive circle packing visualization.

3 CONSIDERATIONS

As we approached building Offbeat, we defined several risks:

- Users might find the result set overwhelming or overstimulating [11].
- Learning to integrate d3.js with the back-end algorithms in a short time frame is ambitious.
- Time constraint; most of the building blocks of this product (data gathering, front-end, back-end, algorithm development, write-up, etc) will take 4-5 weeks individually. Much of this work will be done concurrently but we only have 8 weeks to complete it all.

On the other hand, we also identified several possible payoffs in pursuing building this product and we believe they outweigh the risks:

- Users find value in the product and enjoy exploring new types of music.
- Learning to integrate these tools and taking an idea from its inception to fruition will be a rewarding learning experience.

Costs were around $11 overall: $0 for Google Cloud storage ($0.02 per GB per month, we have 1-10 GBs, and we get $300 in free credit), $10 for a "test" Spotify premium account and $1 to secure the offbeatapp.com domain.

4 OFFBEAT - METHODOLOGY

Offbeat aims to improve upon existing music exploration tools by exposing users to new types of music, offering users visibility into how recommendations are made and giving users power over the degree to which suggested songs differ from music they usually consume. Our modeling and visualization approaches, largely influenced by works cited in our literature survey, work in tandem to achieve this goal.

4.1 Visualization and User Experience

Offbeat's first page is a form inviting the user to grant the tool permission to access their personal Spotify data. At this step, we leverage the Implicit Grant authentication method which is implemented entirely in JavaScript in the user's browser.

After authenticating, we make a call to the 'Personalization' endpoint of the Spotify Web API to fetch the user's top 50 songs from the past six months. We then take those songs' unique Spotify IDs and hit the 'Tracks' endpoint to fetch information about the songs' audio features. We include measurements of eight audio features in our analysis: acousticness, danceability, energy, speachiness, valence, tempo, loudness and liveness. Every song receives a float value for each feature that indicates the degree to which it is present in the song (tempo is assigned a bpm and loudness is measured in decibels).

We use the audio feature data to create a "listening profile", our innovative approach to giving the user visibility into their listening behavior and control over how recommendations are made. The profile includes a bar chart that helps the user identify features they tend to favor when they consume music. The bars break down the average prevalence of each of the eight audio features across the user's top 50 songs. Hovering over an audio feature reveals its description (Figure 1).

The profile also includes a slider the user can manipulate to determine the degree to which Offbeat's suggestions differ from music the user typically consumes (0 being least divergent and 10 being most divergent). This portion of the listening profile addresses our goal of giving the user control over how song suggestions are made. Finally, the user will click the "Explore" button to send a request to our server, which dynamically returns a JSON hierarchy of song data that we use to generate a circle-packing cluster visualization of songs.

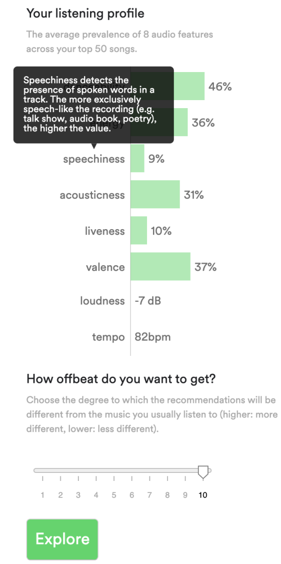


Figure 1: The user's listening profile.

We initially drop the user into to the cluster that corresponds in dissimilarity from their listening profile to the value the user set with the slider (Figure 2). The user can click on small blue song circles to play samples of those songs, and can move between clusters by clicking around the circle packing graph. When moving between clusters, the user is presented with a text overlay on each cluster describing the ways in which its contents differ from adjacent clusters (Figure 3). This provides a level of transparency to our model, allowing the user to understand how songs are grouped and decide which cluster they want to explore based on its audio feature characteristics.

Figure 2: User's initial view with the sample player open for one song.

The degree to which songs differ from what the user typically consumes is indicated by the color of the song circle; dark blue indicates highly divergent songs while light blue/white indicates less divergence. This color mapping is indicated by a color scale at the top of the visualization (Figure 4). The colors of the circle packing visualization were chosen in order to draw users' attention to the song circles. Initial Offbeat implementations contained multiple shades of green in different parts of the circle packing graph that, while aligning with the Spotify branding guidelines, introduced too much visual noise and didn't tell the user a meaningful story (Appendix A.1). Therefore, a light gray was selected as the background, with the only other colors indicating song differentiation and the outer circle's boundary.

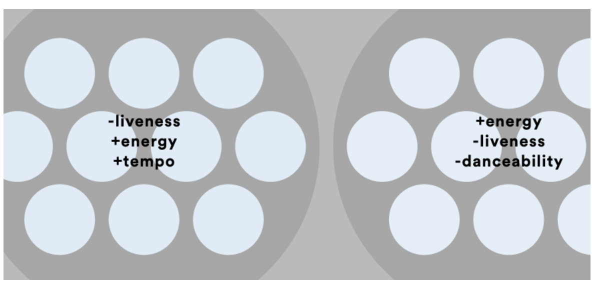


Figure 3: Text overlay on clusters.

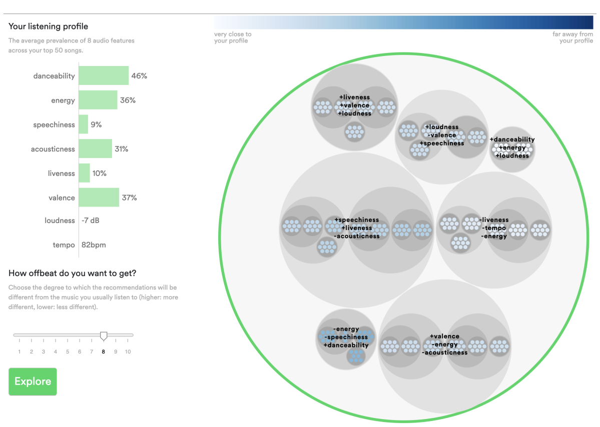


Figure 4: A fully zoomed out view of Offbeat.

4.2 Modeling

As previously stated, Offbeat leverages agglomerative clustering, a bottom-up form of hierarchical clustering. At every iteration under agglomerative clustering, each cluster merges with its least dissimilar neighboring cluster. We calculate cluster dissimilarity using Complete Linkage distance, and we measure dissimilarity of data points with the Euclidean distance.

We apply our model to a data set of three million songs and their audio features (obtained through web scraping and Spotify's Web API). Once clustering is complete, we pass in the user's top 50 songs data and assign the user to a cluster that corresponds in dissimilarity level to the value the user set with the slider seen in Figure 1. This cluster is where the user is initially dropped in the visualization. So although we haven't improved upon the algorithm itself, our modeling approach is innovative in that the cluster assignment is determined based on a user-defined level of dissimilarity from the user's listening profile.

After initial testing of the agglomerative clustering algorithm led to unexpected song groupings, we discovered that the song data is very densely clustered in some areas of the feature space while being sparsely clustered in other areas. We verified this visually using 2-dimensional and 3-dimensional principal component transforms of the data. We dealt with the differences in cluster densities by performing feature selection on the entire songs data set and using a quantile transform function on the remaining features, targeting a Gaussian distribution.

5 EVALUATION

After completing front-end work and back-end development and securing a domain for Offbeat, we set up usability testing to assess how successful we were in our innovative approach to helping users discover new types of music.

5.1 Experiments

We distributed the Offbeat URL and a short description of what the tool aims to accomplish, to friends, family and coworkers. The only variable that factored into our decision of who to include as a tester was whether or not the person has a

Spotify account (since that is a requirement for Offbeat to function). After experimenting with Offbeat, users were encouraged to complete a short survey to provide feedback on the tool. Survey questions were structured to provide answers to the following questions:

- Who are our testers (what does their music consumption look like)?
- How successful was Offbeat in helping testers discover new/different music?
- How effective were the UI/visual elements in making the tester feel they had control over the types of songs that were suggested and insight into how recommendations were made?

Examples of our survey questions corresponding to these categories include:

- In an average day, how many hours of music do you listen to?
- Did this tool help you discover music you've never heard before or don't usually listen to?
- How much control do you feel you had over the types of music the tool recommended?

In total, 33 people tested Offbeat and provided feedback via our survey. As outlined in the following section, we leverage those survey results to better understand how Offbeat performs from a user's perspective and what changes we could make to improve the tool. A full list of survey questions and answers is included in Appendix A.2.

5.2 Findings

The first question our experiment and survey answered was, **who are our testers?** Before analyzing their feedback on Offbeat, we wanted to get a high-level understanding of what our testers' music consumption looks like. We were particularly interested in whether they ever experience the feeling that their music all starts to sound the same, which is one of the main problems Offbeat addresses. In response to the question, *"How often do you feel stuck in a musical rut (i.e. all of your music starts to sound the same and/or you get bored with music recommended to you on streaming platforms)?"* 27 testers said they experience this feeling either sometimes or most of the time, and six said they feel this way either infrequently or never. Therefore, it is clear that this idea of falling into a musical rut resonates with the majority of the testers. Among our test population, Spotify is the most frequently used streaming platform, with Youtube, iTunes and Pandora as the next most popular. More than 60 percent of our testers listen to between two and four hours of music in an average day, with 15 percent listening to more than four hours and 21 percent listening to less than one hour. Given these answers, we can assume the following about our set of testers:

- On average, they listen to between 2-4 hours of music every day
- Most testers use Spotify as their primary streaming platform
- The majority of testers claim to have at some point felt bored with, or limited by, the music being recommended to them on streaming platforms

Having contextualized our testers in terms of their music consumption behaviors, we next wanted to understand **how successful Offbeat was in helping testers discover new music.** This was, after all, our first goal in providing an innovative alternative to traditional recommendation engines. Our survey results were encouraging in that 90 percent of testers said Offbeat helped them discover new music they had never heard before or don't usually listen to. One tester's feedback stated, *"At first I was confused why I had never heard of any of the music offered to me, but then I realized the point of the app was to expose me to music I had never heard before. That being said it did a great job of exposing me to new music I had not yet discovered."* Some testers provided feedback that the music they were recommended was *too* divergent from what they usually consume. This sentiment speaks to the core struggle that traditional recommendation engines face: they must recommend songs that are novel enough to keep the user intrigued, but familiar enough to ensure the user will actually listen to the song suggestions. Offbeat can end up recommending content that is so vastly different from what a user is accustomed to that the user might simply not want to listen to it. This exemplifies the benefit of Offbeat's slider that allows the user to adjust the level to which the suggested songs stray from what they might enjoy consuming.

The remainder of our survey questions assessed the effectiveness of Offbeat's UI/visual elements. Some of our questions asked about whether testers took basic actions such as clicking around to different song clusters and changing the slider value. All but one user clicked outside of their initial cluster, and all but two changed the slider value. Therefore, almost all users took advantage of Offbeat's interactive elements without explicit instructions on how to do so, suggesting a fairly intuitive design methodology. However, the question we were most interested in answering around our UI elements was **how effective those elements were in making the tester feel they had control over recommendations and insights into how recommendations were made.** The text overlay describing the differences between clusters of songs appears to have been successful in helping testers understand how recommendations were made since 98 percent of testers said they at least somewhat understood the ways in which groups of recommendations differed from each other. However, when testers were asked how much control they felt they had over the types of music Offbeat recommended on a scale of one to five (one being no control and five being complete control), the average response was 2.3. In the short answer responses, many testers mentioned they would have liked to have had more control over which audio features or what genres would appear in their suggested songs. So although Offbeat technically provides the user more control over song suggestions than traditional recommendation engines, it could still provide more opportunities for users to home in on specific elements that are important to them.

6 CONCLUSION

This project helped our group grow in many ways as we had to quickly learn to integrate multiple moving parts in order to provide users with a working product. With all group members contributing similar amounts of effort to build Offbeat, we sharpened our skills in project management and teamwork, UI design, d3.js, web scraping, cloud deployment and modeling/clustering.

Our goal in building Offbeat was threefold: to help users discover songs that are dissimilar to what they typically consume, to give those users visibility into how recommendations are made, and to give the user control over what those recommendations look like. From our usability testing, we found that the current Offbeat implementation is successful in the first two objectives while it has room to grow in offering users control over the types of songs that are suggested. User feedback also revealed it would be helpful if we provided additional direction within the tool around how to navigate between clusters as it wasn't immediately intuitive to all users that clicking outside of their initial cluster would allow them to explore the entire visualization. Although there are many improvements we could make if we were to continue developing this tool, the user feedback on Offbeat's concept and its execution was overall positive, with many users providing feedback like, *"As a tool it's incredibly interesting to think about what is the 'opposite' of what I normally listen to,"* and, *"I spent much more time on it [Offbeat] than I expected...there's clearly potential in this idea."* With about 90 percent of testers saying they would use a tool like this (that highlights music dissimilar to what they usually consume), we believe this concept is something music consumers at large would be excited about.

REFERENCES

[1] Stephan Baumann and Oliver Hummel. 2005. Enhancing music recommendation algorithms using cultural metadata. *Journal of New Music Research* 34, 2 (2005), 161–172.

[2] Jinpeng Chen, Pinguang Ying, and Ming Zou. 2019. Improving music recommendation by incorporating social influence. *Multimedia Tools and Applications* 78, 3 (2019), 2667–2687.

[3] R Elmasri and S Navathe. 2016. Fundamentals of Database Systems. 7. kansainvälinen p.

[4] Craig Hamilton. 2019. Popular music, digital technologies and data analysis: New methods and questions. *Convergence* 25, 2 (2019), 225–240.

[5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2017. *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media.

[6] Sofia Johansson, Ann Werner, Patrik Åker, and Greg Goldenzwaig. 2017. *Streaming music: Practices, media, cultures.* Routledge.

[7] Gérald Kembellec, Ghislaine Chartron, and Imad Saleh. 2014. *Recommender systems.* Wiley Online Library.

[8] Madjid Khalilian, Norwati Mustapha, and Nasir Sulaiman. 2016. Data stream clustering by divide and conquer approach based on vector model. *Journal of Big Data* 3, 1 (2016), 1.

[9] Jeremy Wade Morris. 2015. Curation by code: Infomediaries and the data mining of taste. *European Journal of Cultural Studies* 18, 4-5 (2015), 446–463.

[10] Jeremy Wade Morris and Devon Powers. 2015. Control, curation and musical experience in streaming music services. *Creative Industries Journal* 8, 2 (2015), 106–122.

[11] Lillian Pierson and Jake Porway. 2017. *Data science for dummies.* John Wiley & Sons.

[12] Erin Renshaw and John Platt. 2009. Client-based generation of music playlists via clustering of music similarity vectors. US Patent 7,571,183.

[13] Swizec Teller. 2013. *Data Visualization with d3. js.* Packt Publishing Ltd.

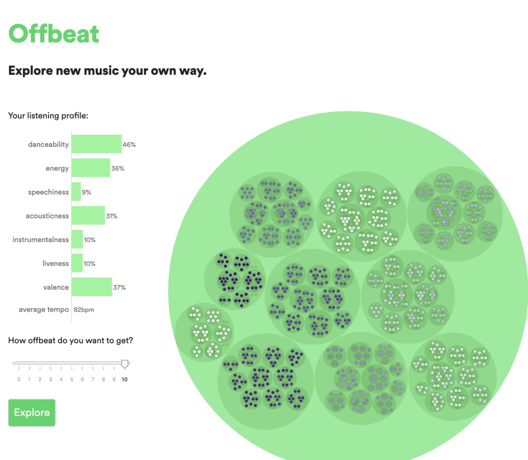
[14] Stephen A Thomas. 2015. *Data visualization with javascript.* no starch press.

[15] Wei-Ho Tsai, Dwight Rodgers, and Hsin-Min Wang. 2004. Blind clustering of popular music recordings based on singer voice characteristics. *Computer Music Journal* 28, 3 (2004), 68–78.

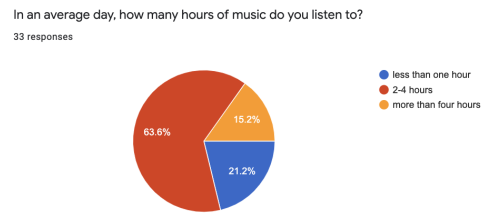
[16] Ranran Wang, Xiao Ma, Chi Jiang, Yi Ye, and Yin Zhang. 2020. Heterogeneous information network-based music recommendation system in mobile networks. *Computer Communications* 150 (2020), 429–437.

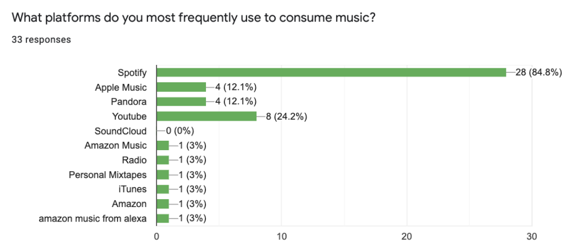
A APPENDIX

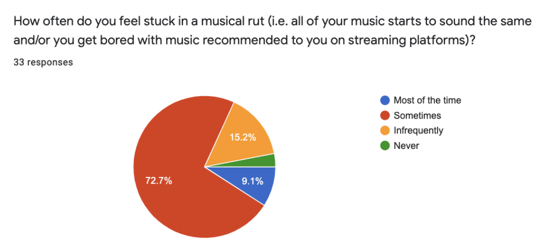
A.1 Initial Offbeat Color Palette

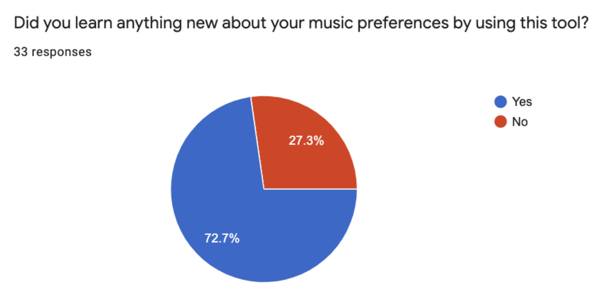


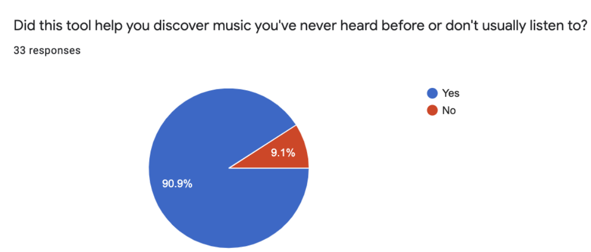
A.2 Survey Questions and Answers





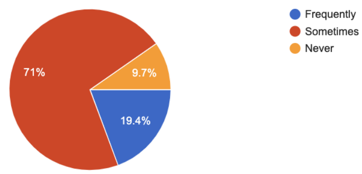






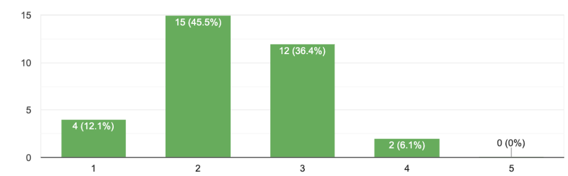
How often would you use a tool like this (one that highlight songs dissimilar to those you usually consume) to find new music?

31 responses



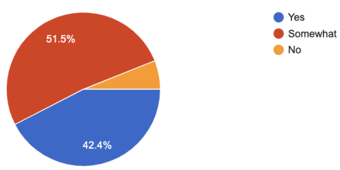
How much control do you feel you had over the types of music the tool recommended?

33 responses



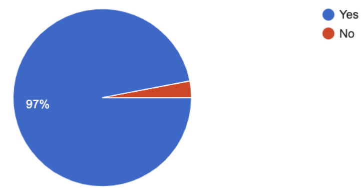
Did you understand the ways in which groups of song recommendations differed from each other and from the music you usually consume?

33 responses



Did you navigate around the visualization (click in and out of groups of circles)?

33 responses



Did you change the "Offbeat" slider value before clicking the "Explore" button?

33 responses