Predicting Song Preference

About the Data

Services such as Spotify and Pandora offer a radio-like experience of songs based on user preference. Using the same metrics, can we provide and build more relevant and personalized recommendations?

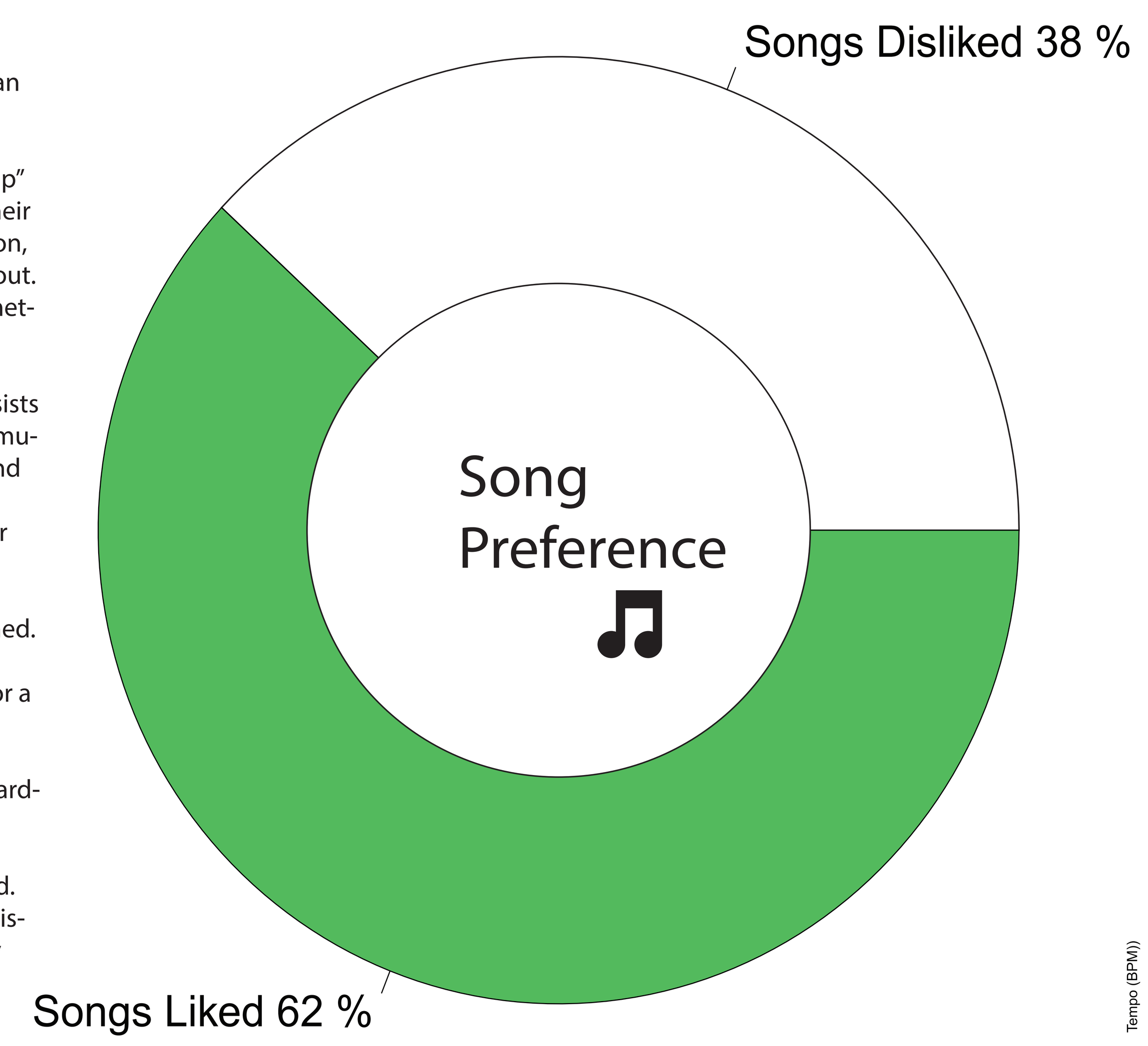
One limitation of Spotify is that the service only gets feedback about a user's preference if the user explicitely selects "thumbs up" or "thumbs down" while listening to a track. For many reasons, a user may be unable to select this as they could be away from their device, not be able to make a selection before the song finishes, or simply be unwilling. Without this steady stream of information, Spotify can be biased in that users may partake in a voter bias in which they only vote for or against songs they feel strongly about. It would be interesting to compare song preference/profiling based on the Spotify system in it's current state versus the same metrics with explicit preference annotations.

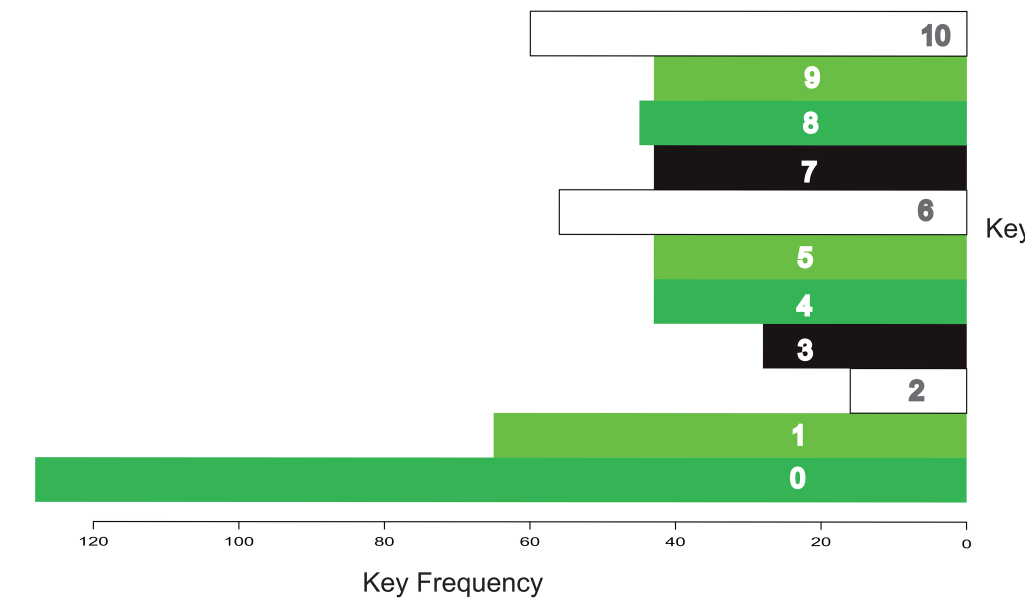
This dataset comes from a user by the name of George McIntire on Kaggle: the dataset is titled Spotify Song Attributes and consists of 16 columns and 2017 rows. Most of the attributes in this dataset are from Spotify's API which returns a set of Spotify-specific musical metrics such as "Danceability" and "Energy". These metrics are unique to Spotify and tend to be a scaled value between 0 and 1. More information regarding the Spotify values can be found in the API documentation at Spotify For Developers or developer.spotify.com. Another column provided by George is the target column which designates whether George liked the song (0) or not (1).

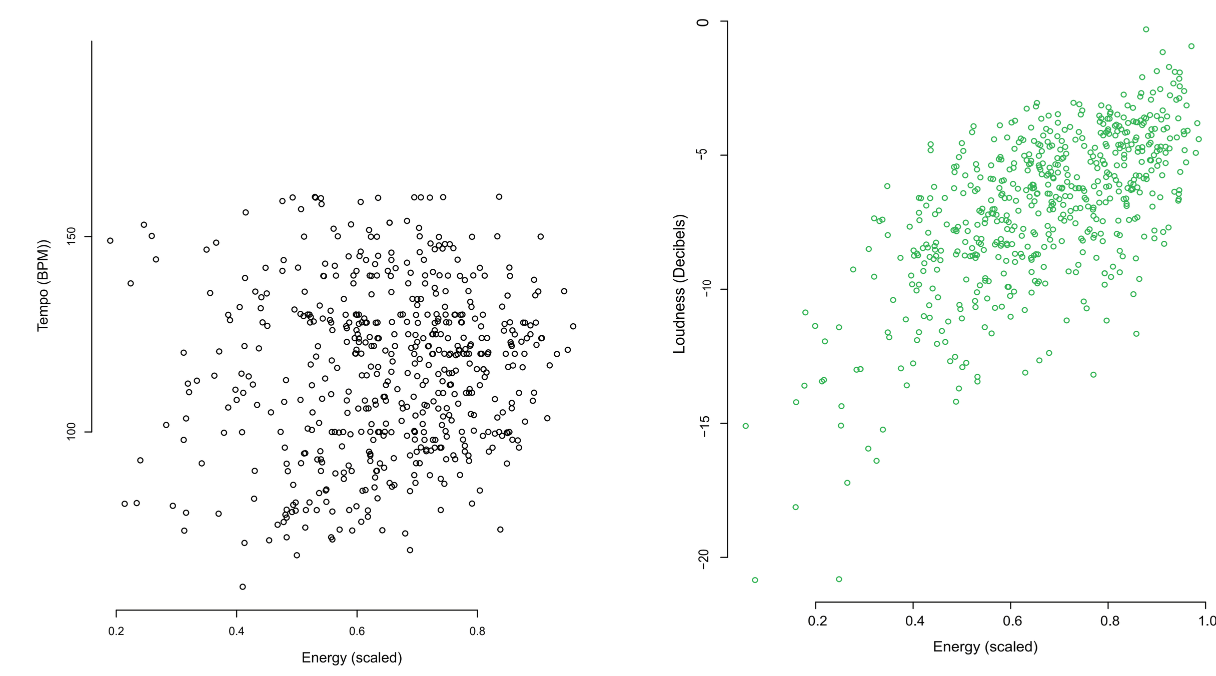
Unfortunately, while the Spotify API provides a wide range of quantitative metrics, it does not provide the genre of songs returned. I decided to downsize the original dataset of 2,017 rows to 670; I added the genre information myself by referencing Music Billboards and Wikipedia. As a comparison as well as validation method, I also submitted the dataset to Amazon Mechanical Turk for a second annotation. I then calculated Cohen's kappa value, which is a metric that describes how much agreement was due to chance. Cohen's kappa between my annotations and the AMT annotations is -0.042 which means that there is hardly any agreement with the two. The AMT annotators were allowed to use other resources to answer these questions but as mine were standardized and from reputable sources, I used my annotations as the default genre labels.

Looking at the distribution of liked songs to disliked songs, the sample has a stronger representation of liked songs than disliked. Many of the songs are also in the 0 key — according to the API documentation, this is correlated to the key C major. This is surprising because Spotify's own Data Engineer Kenny Ning had released an infographic stating that amongst all records in the Spotify database — G major is the most common.

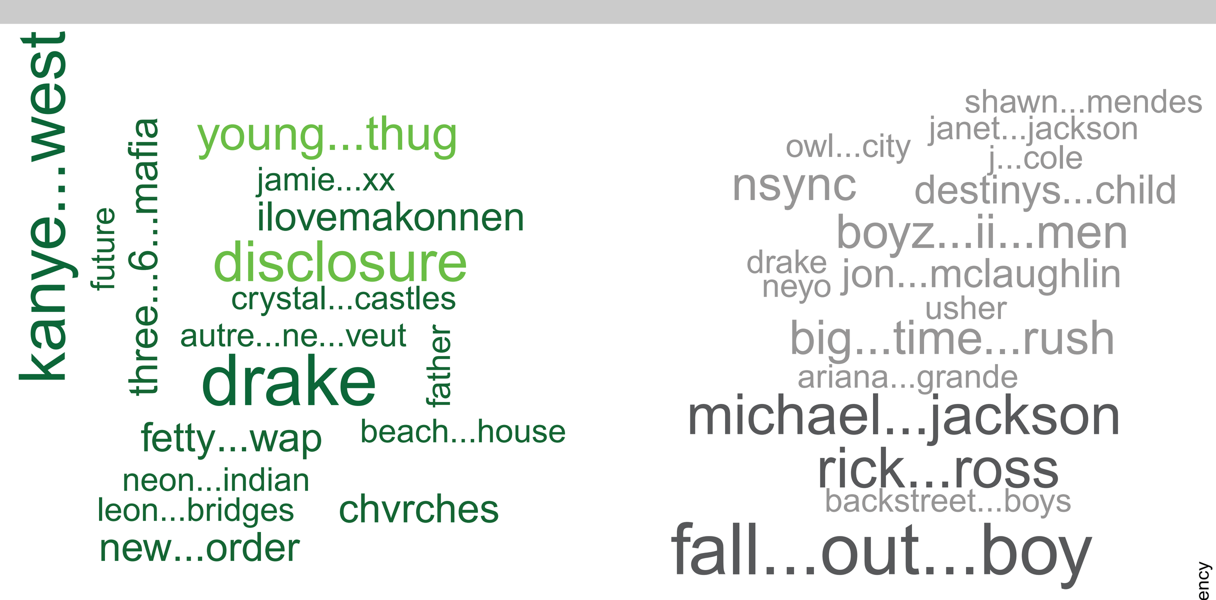
Another interesting thing to mention about the Spotify metrics is that at least two of the metrics are functions of other metrics. Energy is a value calculated using Loudness, entropy, and timbre, while Danceability Is based on Tempo and rhythm. The association between Energy and Loudness appears to be a much stronger positive correlation while the association between Danceability and Tempo is weaker and is like more strongly influenced by other factors.





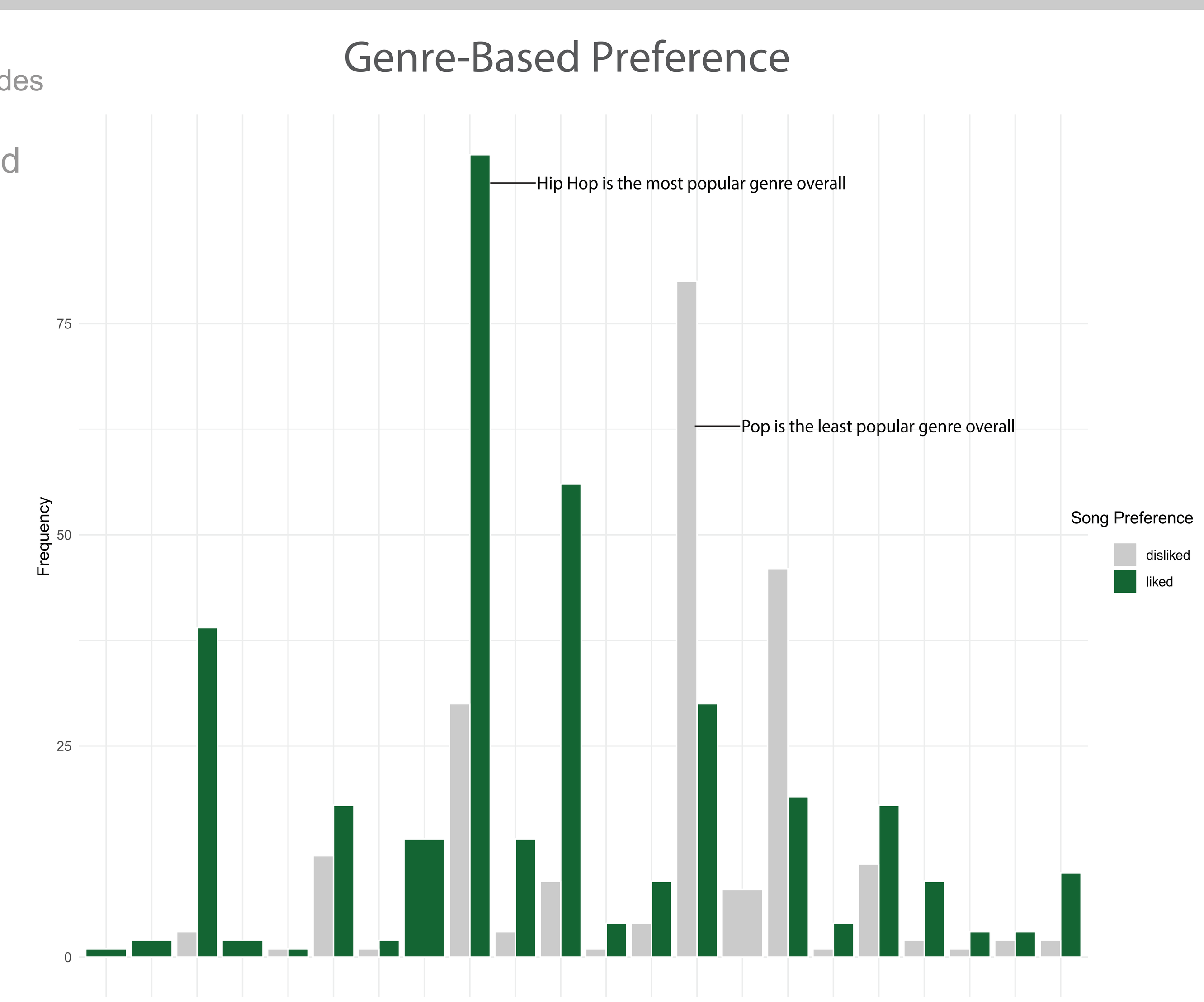


Questions

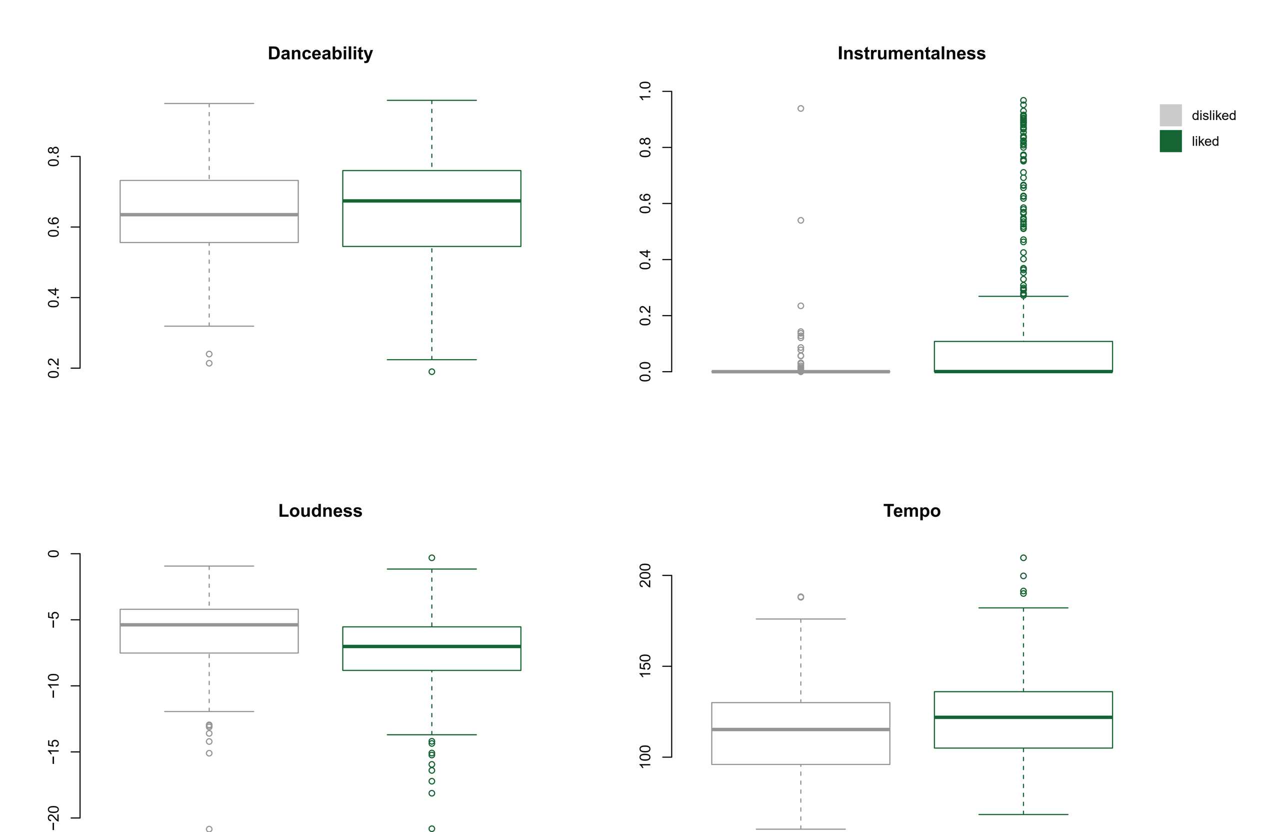


Is there an artist bias?
The author of the dataset seems to have a preference for Hip Hop artists like Kanye West and Drake. He also seems to dislike Pop artists like Fall Out Boy and the Backstreet Boys. It's interesting that Drake appears in both the liked artists word cloud as well as the disliked artist cloud; this is because the author prefers the artist's Hip Hop and Trap songs and dislikes his Pop singles.

Is there a genre bias?
As expected, the author really likes Hip Hop songs and dislikes Pop songs. The author likes all Disco, and classical songs in the set and dislikes all Punk songs in the set.

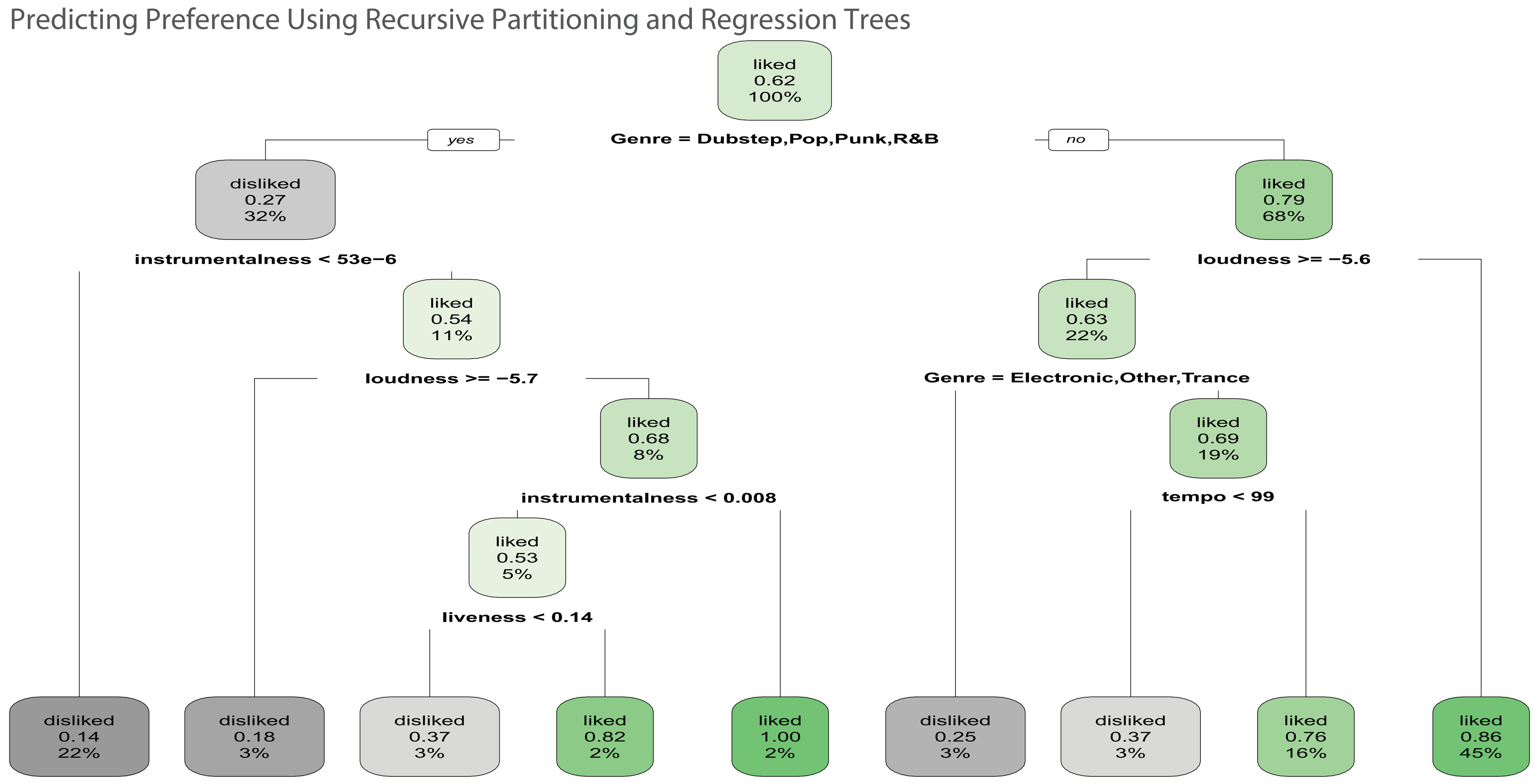


What is the makeup of liked/disliked songs?
the author prefers songs with higher danceability, higher instrumentalness, lower loudness, and higher tempo.



Predictions

Predicting Preference Using Recursive Partitioning and Regression Trees



Can we predict liked songs?
Yes! Based on genre, loudness, instrumentalness, liveness, and tempo; we can determine if the user will like songs or not. The user tends to like songs that are not dubstep, pop, punk, or re&b that aren't very loud. The user does not like songs with low instrumentalness, loud songs, or live songs. The accuracy, or percent of the data correctly represented by this model is 58.97.%

Sources:
"Get Audio Features for a Track." Spotify for Developers, https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/.
McIntire, George. "Spotify Song Attributes." Kaggle, 4 Aug. 2017, https://www.kaggle.com/geomack/spotifyclassification/data.
Misra, Ria. "A Chart Of The Most Commonly Used Keys Shows Our Actual Musical Tastes." io9, io9, 16 Dec. 2015, https://io9.gizmodo.com/a-chart-of-the-most-commonly-used-keys-shows-our-actual-1703086174.
"Rpart." Function | R Documentation, https://www.rdocumentation.org/packages/rpart/versions/4.1-15/topics/rpart.

Conclusion

In conclusion, the author has very clear biases towards certain genres and certain artists. We can see that the author prefers Hip Hop to Pop songs and is partial to Kanye West and Drake. We can create and visualize decision trees to see the flow of the author's song preference. The author tends to dislike songs that have higher instrumentalness (more instrumental than vocal), loud songs, and live songs. Products like Spotify only work if given an appropriate amount of information -- the average user will not list each and every song or artist they like and therefore may be recommended songs they don't like due to lack of information. Users may be interested to see, when given explicit information regarding liked or disliked songs, whether the product will reccomend more relevant suggestions. Competitors can also try to solve the issue of obtaining this customer information without asking the user to explicitely choose "like" or "dislike" continuously which many users will either be unwilling or unable to do so during song play.

Aneeka Latif
IST 719