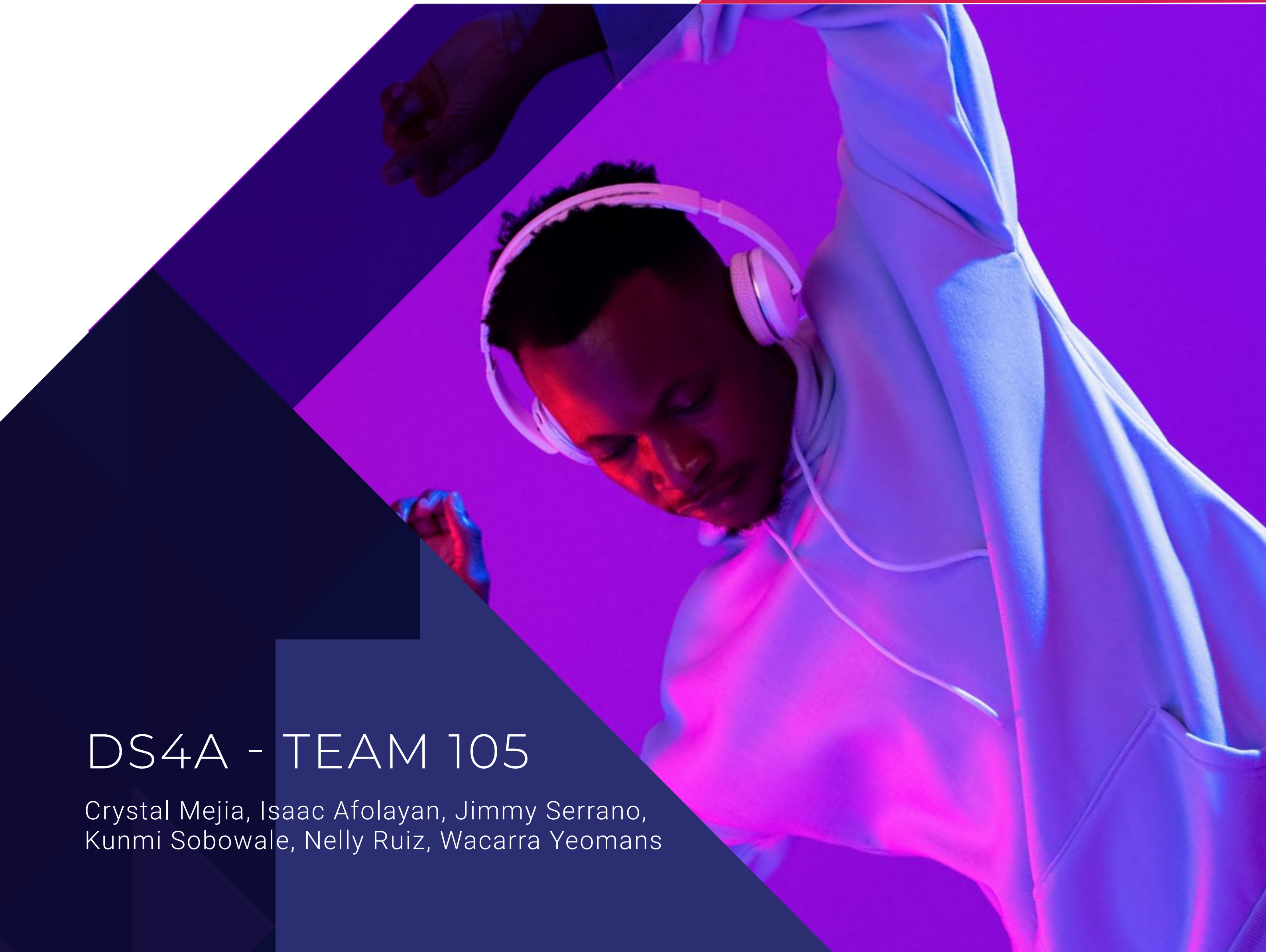
BREAK IT DOWN

CAN AUDIO ATTRIBUTES DETERMINE A HIT SONG?

FINAL REPORT

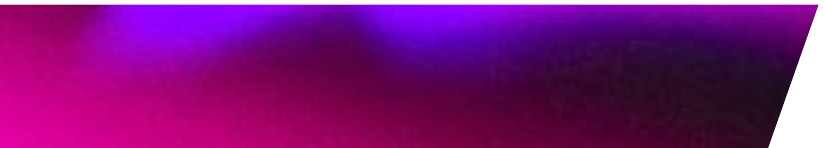
AUG 13, 2021



DS4A - TEAM 105

Crystal Mejia, Isaac Afolayan, Jimmy Serrano, Kunmi Sobowale, Nelly Ruiz, Wacarra Yeomans

"THE TRUE BEAUTY OF MUSIC IS THAT IT CONNECTS PEOPLE." - RAY AYERS



CONTENTS

01. INTRODUCTION

Business Problem

According to the Recording Industry Association of America (RIAA), the rise in music streaming has allowed more songs than ever before to earn gold or platinum awards[1]. In addition, music streaming platforms, such as Spotify, have allowed the music industry to move to digital and have changed the market prioritization. In the last few years, the music industry has been focusing more heavily on singles over albums. Singles require less effort and preparation, have more potential to reach viral status, and open new opportunities to a broader range of artists. In this new music ecosystem, artists and labels are challenged to invest their time, energy, and resources to create music that will reach a broad audience.

With this in mind, our team is interested in understanding patterns and features in popular music in recent years. Being successful in the music industry is a difficult task for many artists and record labels, so our project aims to use data science techniques to understand the makings of a popular song better. In addition, our project will determine which song features help determine whether a song will be a hit or not on both Spotify and Billboard.

Business Impact

We see an opportunity to provide record labels and artists with insights on the features of popular music that will assist in their creation process. Our solution examines what combinations of song features (audio features, artist profile, label profile, etc.) determine song popularity on Spotify and Billboard. Ultimately, our project allows different stakeholders in the song production process to make informed, data-driven decisions.

MUSIC INDUSTRY BY THE NUMBERS[2]

$5B Anticipated revenue from streaming by

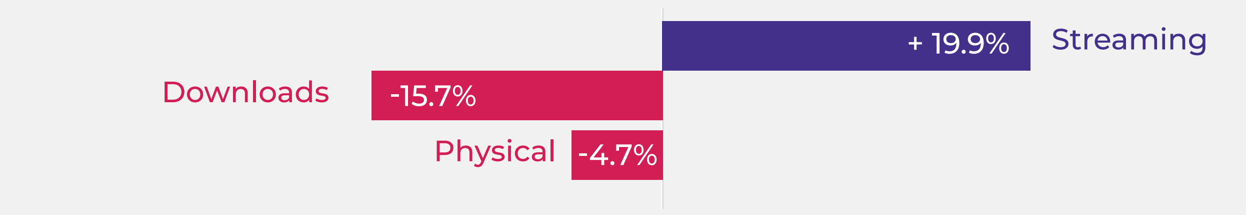
$2M Cost to break a new artist into a major market

4 US Consumers spend an average of **HOURS/DAY** listening to musc

[1] Hissong, Samantha. "More Songs Are Going Platinum Than Ever Before." Rolling Stone

[2] Source: CompareCamp 73 Music Industry Statistics 2021

2020 Music Industry Growth/Decline by Format



Why It Matters

While other existing projects explore the song features of popular music, our project also explores the relationship between songs that Spotify deems popular versus Billboard's End of Year charts. Streams determine popularity on Spotify. However, Billboard's End of Year chart is a reflection of a song's commercial success. By identifying the features that make a song a hit on Spotify and Billboard, we can help inform the song creation process or launch prioritization. As a result, artists and labels have songs that reach a wider audience and generate more revenue.

> *It was a strong year for streaming and revenues grew by 19.9% in 2020 to US$13.4 billion. Paid subscription streaming was the key driver of this, growing 18.5%.*
>
> Global Music Market Overview 2020

02

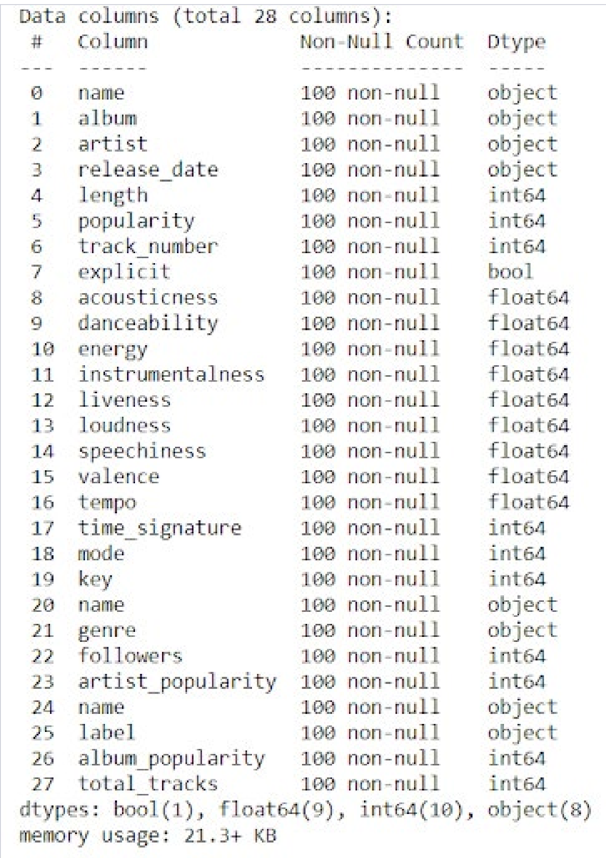
DATA ANALYSIS & COMPUTATION

2.A. DATA SETS & DATA CLEANING

We used the Spotify and Billboard APIs to gather data about tracks for 2017-2020 and build our primary dataset for our project.

Billboard

Billboard is a US-based magazine and website that produces news and reviews related to the music industry. The Billboard Year-End charts are a cumulative measure of a single or album's performance in the US. The measurements for this list are a total of yearlong sales, streaming, and airplay points, and it runs from the first week of December to the final week in November[2]. We used Billboard.py, a Python library, to extract data about the top songs for 2017-2020 in the United States.



```
Data columns (total 28 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   name                100 non-null    object
 1   album               100 non-null    object
 2   artist              100 non-null    object
 3   release_date        100 non-null    object
 4   length              100 non-null    int64
 5   popularity          100 non-null    int64
 6   track_number        100 non-null    int64
 7   explicit            100 non-null    bool
 8   acousticness        100 non-null    float64
 9   danceability        100 non-null    float64
 10  energy              100 non-null    float64
 11  instrumentalness    100 non-null    float64
 12  liveness            100 non-null    float64
 13  loudness            100 non-null    float64
 14  speechiness         100 non-null    float64
 15  valence             100 non-null    float64
 16  tempo               100 non-null    float64
 17  time_signature      100 non-null    int64
 18  mode                100 non-null    int64
 19  key                 100 non-null    int64
 20  name                100 non-null    object
 21  genre               100 non-null    object
 22  followers           100 non-null    int64
 23  artist_popularity   100 non-null    int64
 24  name                100 non-null    object
 25  label               100 non-null    object
 26  album_popularity    100 non-null    int64
 27  total_tracks        100 non-null    int64
dtypes: bool(1), float64(9), int64(10), object(8)
memory usage: 21.3+ KB
```

The initial data obtained from the Spotify API required some reworking to suit our analysis needs. The top tracks for 2017-2020 were saved as individual CSVs given the 100 track limit through the Spotify API. Each separate CSV contained 28 columns.

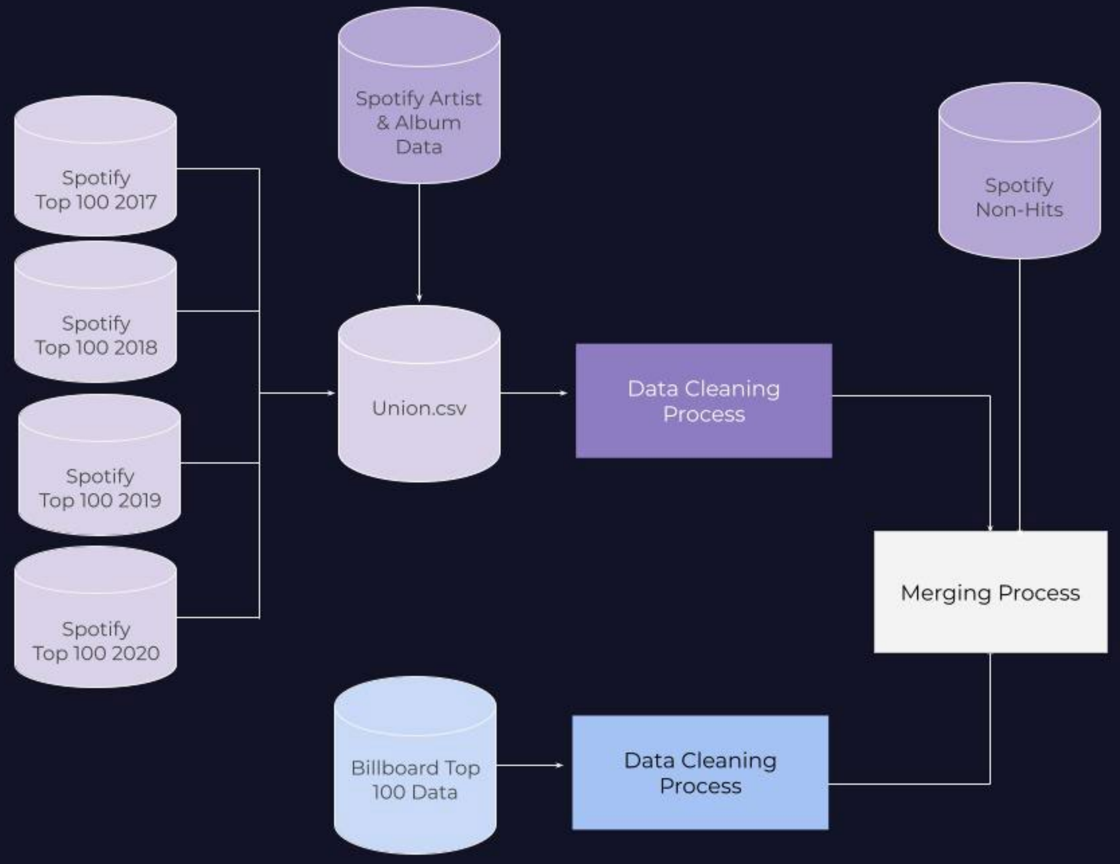
Spotify

Spotify is an audio streaming service that operates in 178 countries and has accumulated a library of over 70 million songs. Users can search for music based on artist, album, or genre and create, edit, and share playlists. In addition to user-generated playlists, Spotify also provides users access to playlists curated by the platform itself. Every year, Spotify releases a Top Tracks playlist that features the top global tracks for that specific year.

For our project, we used Spotipy, a Python library, to access the Spotify API and retrieve data on the audio, artist, and album features for the top songs of 2017-2020 based on Spotify's Top Tracks playlists. The playlists for 2017 and 2018 had 98 and 100 top tracks, respectively, while the 2019 and 2020 playlists had 50 top tracks.

Along with the top tracks for 2017-2020, we also collected data on non-hit songs released by the top major labels (Universal Music Group, Sony Music Entertainment, Warner Music Group, and EMI)[3]. Again, our team used a selection of 500 songs each year and each label to create a dataset of non-hit songs. Furthermore, as with the top tracks, we obtained the same audio, artist, and album feature categories for the non-hits.

CLEANING PROCESS



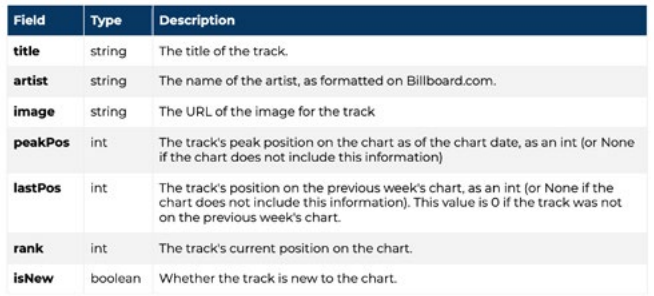
Using the Pandas library, we combined all the individual CSVs into a master dataframe and used the following data cleaning process:

Hits

1. The CSVs were saved on our GitHub repository and imported into our Jupyter notebooks using the requests library in Python.
2. Created the SpotifyList column to identify which playlist year the song appeared.
3. Removed the first column because it was a duplicate of the index and the duplicate name column
4. Converted the track length data from milliseconds to minutes
5. Added a new column (length (sec)) that converts track length from minutes to seconds
6. Converted release_date into DateTime data type and parsed it into month, year, weekdayName and made these new column object types
7. Checked for the presence of NA values
8. Since the data had particular genre categories from Spotify, we created dictionaries to recategorize the genres into major genres, like pop, rap, or hip-hop. We recategorized the label column into the parent company and the specific sub-label
9. Created a new column for the rank of the tracks that were on Spotify playlists
10. Confirmed the data types found in the dataframe (.info)

Non-Hits

The dataset of non-hit songs released by the major labels also went through a similar process as described previously. One of the additional primary steps required for the non-hit song dataset included finding the unique label names and removing the tracks that belonged to regional/local subsets of the major labels (e.g. Sony Canada). We did this step, so the dataset was reflective of a broad/global audience. Once clean, We merged the top tracks and non-hit tracks.

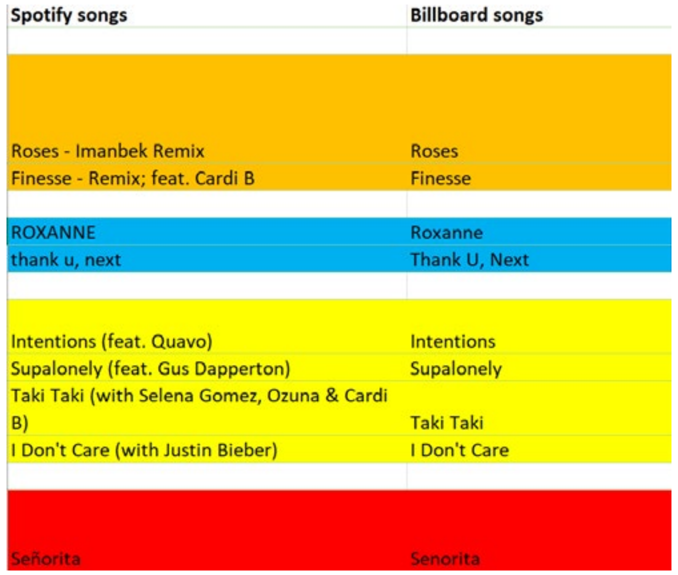


Field	Type	Description
title	string	The title of the track.
artist	string	The name of the artist, as formatted on Billboard.com.
image	string	The URL of the image for the track
peakPos	int	The track's peak position on the chart as of the chart date, as an int (or None if the chart does not include this information)
lastPos	int	The track's position on the previous week's chart, as an int (or None if the chart does not include this information). This value is 0 if the track was not on the previous week's chart.
rank	int	The track's current position on the chart.
isNew	boolean	Whether the track is new to the chart.

For the Billboard data, the API provided data that needed to be exported to a CSV file and then re-uploaded before using the Pandas library for data cleaning. We followed the following steps the data cleaning process:

1. Transposed the CSV row to column
2. Reset index to make a column containing artists and song names and renamed it to combined
3. Converted combined from object to string
4. Split combined into artist and song columns
5. Removed the apostrophe character from the song name
6. Removed featured artist information from the artist column

MERGING PROCESS



Spotify songs	Billboard songs
Roses - Imanbek Remix	Roses
Finesse - Remix; feat. Cardi B	Finesse
ROXANNE	Roxanne
thank u, next	Thank U, Next
Intentions (feat. Quavo)	Intentions
Supalonely (feat. Gus Dapperton)	Supalonely
Taki Taki (with Selena Gomez, Ozuna & Cardi B)	Taki Taki
I Don't Care (with Justin Bieber)	I Don't Care
Señorita	Senorita

To create one final dataset for use, we cleaned the song titles in the Billboard and Spotify datasets to facilitate the matching process. First, the process involved removing featured artists from song titles and artist names. Next, we made all titles lowercase. See the image on the left for an example of the title differences between the Spotify and Billboard song titles.

Using the fuzzywuzzy library, we conducted fuzzy matching using the Levenshtein Distance score cutoff of 85% to specify if the song titles between the two datasets were a match or not. The fuzzy matching results were merged to the Spotify dataset using an inner join. The Billboard dataset was then merged to the Spotify dataset using an outer join.

FINAL DATASET

Field	Type	Description[4]
rank	int64	Ranking calculated from total streams on Spotify for a given year.
track	object	The title of the track.
album	object	The album the title is featured on.
artist	object	The name of the artist.
release_date	object	The release date of the track (YYYY-MM-DD).
length	int64	The duration of the track in minutes.
popularity	int64	The popularity of the track, based on the time the data was pulled. The value will be between 0 and 100, with 100 being the most popular.
track_number	int64	Track number on the album.
explicit	bool	Whether or not the track has explicit content (TRUE if it does and FALSE if it does not or it is unknown).
acousticness	float64	A confidence measure from 0.0 to 1.0 of whether the track is acoustic, with 1.0 representing high confidence that the track is acoustic.
danceability	float64	A measure from 0.0 to 1.0 that describes how suitable a track is for dancing based on a combination of musical elements, including tempo, rhythm stability, beat strength, and overall regularity. A value of 1.0 represents the most danceable.
energy	float64	A confidence measure from 0.0 to 1.0 that represents a perceptual measure of intensity and activity.
instrumentalness	float64	A confidence measure from 0.0 to 1.0 that predicts whether a track contains no vocals. The closer the value is to 1.0, the greater likelihood the track contains no vocal content.
liveness	float64	A confidence measure from 0.0 to 1.0 that detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.
loudness	float64	The overall loudness of a track in decibels (dB).Values typically range between -60 and 0 db. Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks.
speechiness	float64	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.
valence	float64	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
tempo	float64	The overall estimated tempo of a track in beats per minute (BPM).
time_signature	int64	An estimated overall time signature of a track. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).
mode	int64	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
key	int64	The key the track is in. Integers map to pitches using standard Pitch Class notation.
SpotifyList	int64	The year the song was on the "Top Track" Spotify list.
month	int64	The month the song was released.
year	int64	Year the song was released.
weekdayName	object	Day of the week the song was released.
genre	object	A list of the sub genres the artist is associated with.
followers	float64	The number of followers an artist had at the time the data was pulled.
artist_popularity	float64	The popularity of the artist, based on the time the data was pulled. The value will be between 0 and 100, with 100 being the most popular. The artist's popularity is calculated from the popularity of all the artist's tracks.
label	object	The record label for the album.
album_popularity	int64	Popularity of the album, based on the time the data was pulled.
total_tracks	int64	Number of tracks on the album containing the hit song.
genre_cat	object	A list of the major genres the artist is associated with.
label_cat	object	The parent music organization for the album.
track_clean	object	Fuzzy matching Spotify song titles with artists.
billboard_song _name	object	Fuzzy matching song titles.
score	int64	Fuzzy matching score based on Levenshtein Distance
billboard_rank	int64	The track's position on the chart, based on the time the data was pulled.
billboard_year	int64	The year the song was on the Billboard chart.
billboard_original_n ame	object	Original title of song on Billboard chart.
which_list	object	Whether the song is on the Spotify or Billboard list, both, or none.

2.B. EXPLORATORY DATA ANALYSIS

Using our dataset of hit and non-hit songs spanning four years, our team explored trends surrounding the genre, record label, rank, release date, and audio attributes. One of our main goals was to investigate whether popular songs changed over the years as a whole or only for specific characteristics. Therefore, additional data frames based on ranking subgroups (e.g., Top 50, Top 25, and Top 10 tracks) were created for our EDA to investigate any notable trends in these smaller clusters of songs.

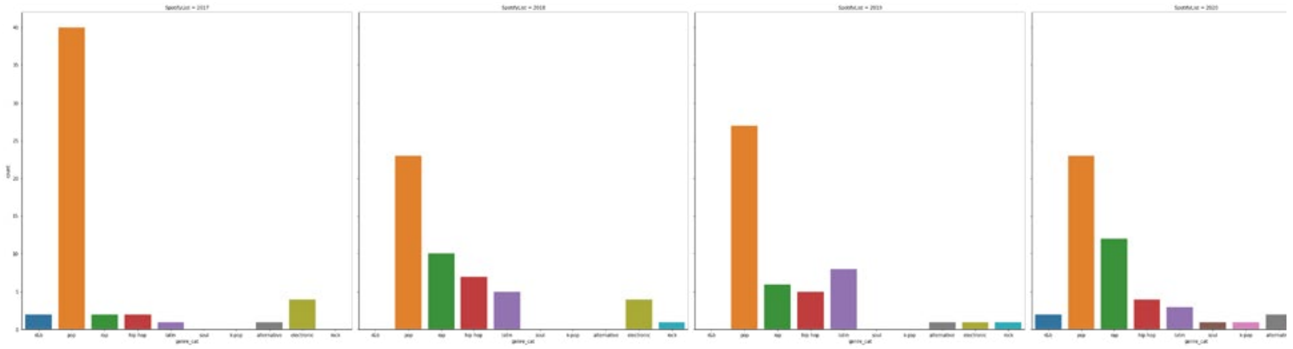
HYPOTHESIS

We theorized that pop music would be the genre with the most popular songs across all years for our EDA. We also hypothesized that a combination of audio attributes would be associated with song popularity/rank based on prior studies. Specifically, we believed that song danceability and energy would be necessary. Finally, we hypothesized that valence, which measures song positivity, might be lower in 2020, given the pandemic and society's overall mood.

OBSERVATIONS FOR TOP TRACKS

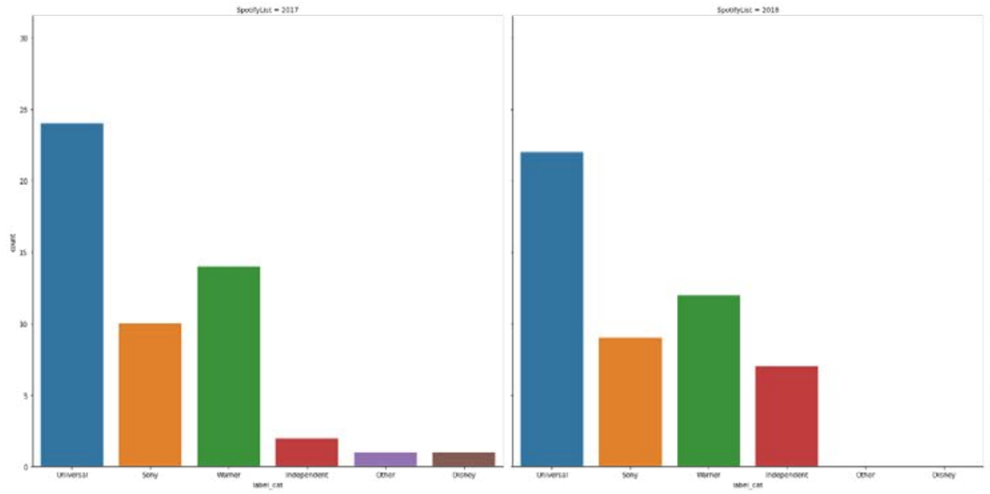
Genres

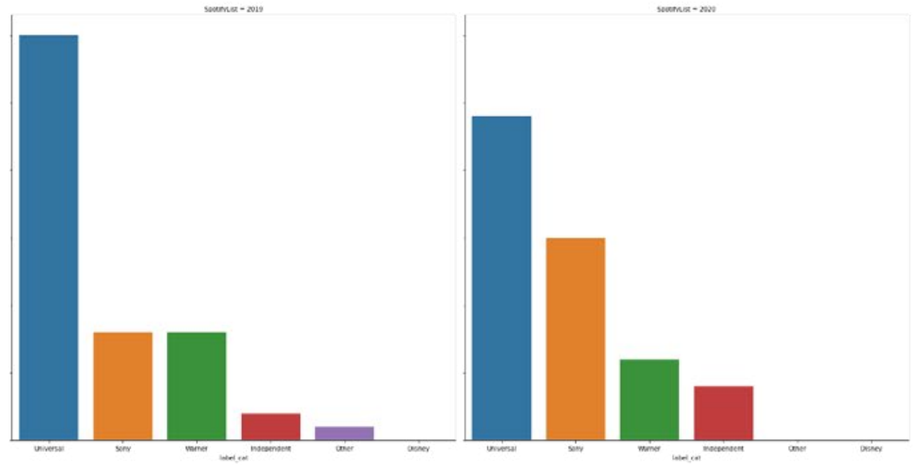
From 2017 to 2020, there have only been ten different genres among the top tracks: r&b, pop, rap, hip hop, Latin, soul, k-pop, alternative, electronic, and rock. As hypothesized, the most popular across all years has been pop. Rap, hip hop, and Latin have been the next most popular, all three experiencing growth throughout the years. However, when filtering for just the Top 10 tracks each year, there is more variability of the genre from year to year. For example, rap tied for most tracks in 2020 and had the most in 2018 in the top 10.



Labels

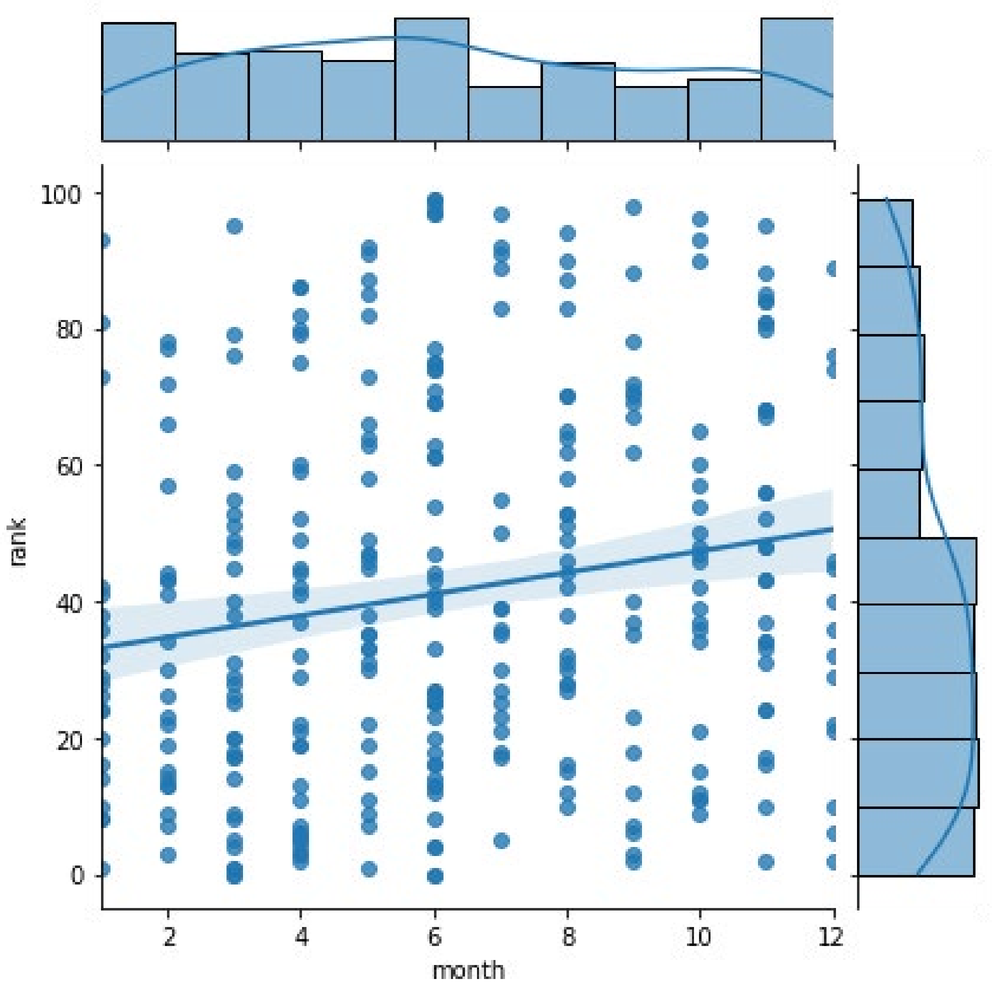
Among the top tracks for 2017 to 2020, there have only been three major labels representing artists: Universal, Sony, and Warner. The other labels include Disney and independent labels. The most popular label across all years has been Universal. When filtering for just the top 10 tracks each year, there is more variability among the labels from year to year. For example, Sony and Universal tied for most tracks in 2017 and 2020 among the top 10 tracks.

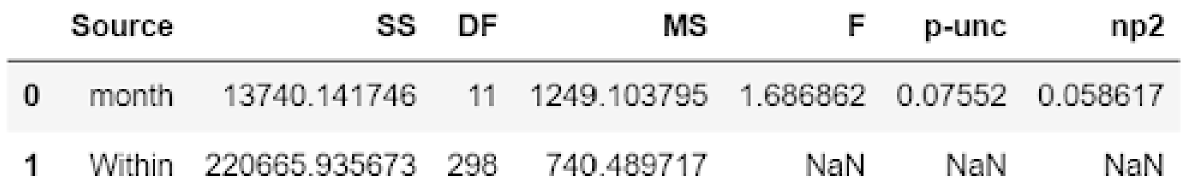




Release Date

A scatter plot of the release month and song rank suggested that earlier release months boosted Spotify rank. However, an ANOVA analysis did not find an association between month and rank.

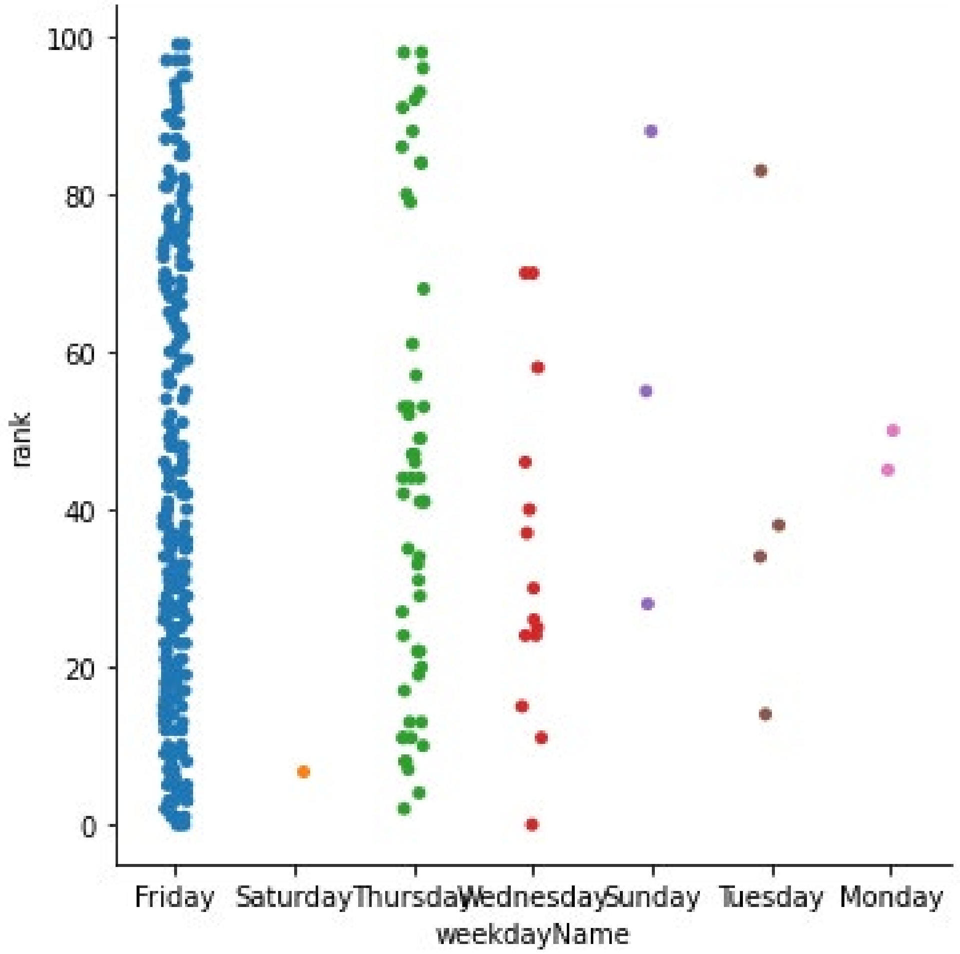


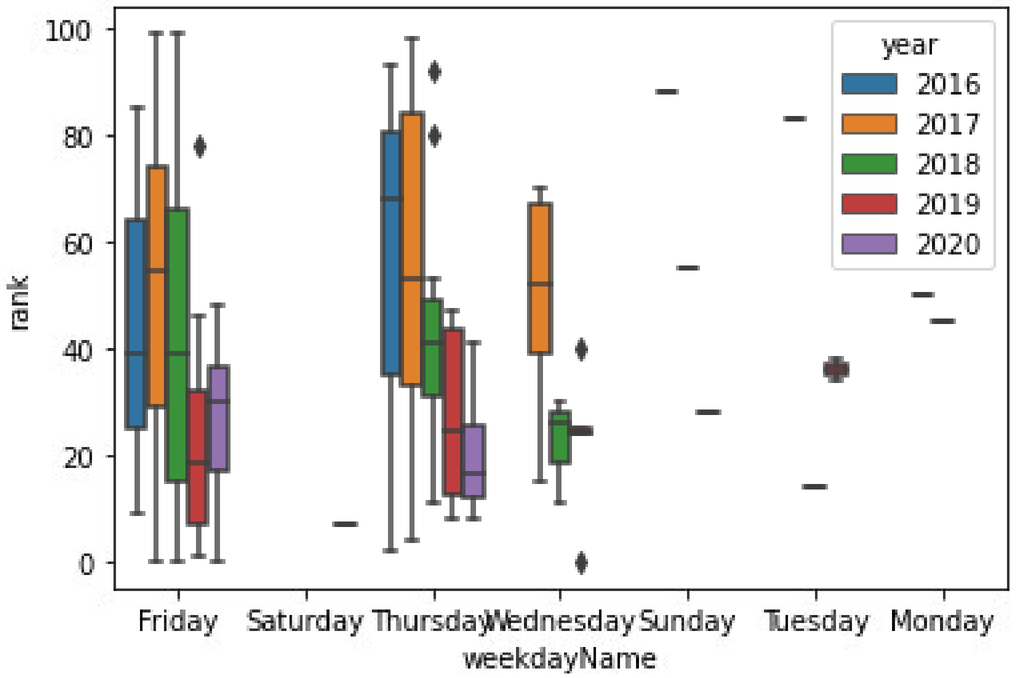


	Source	SS	DF	MS	F	p-unc	np2
0	month	13740.141746	11	1249.103795	1.686862	0.07552	0.058617
1	Within	220665.935673	298	740.489717	NaN	NaN	NaN

Release Date (cont.)

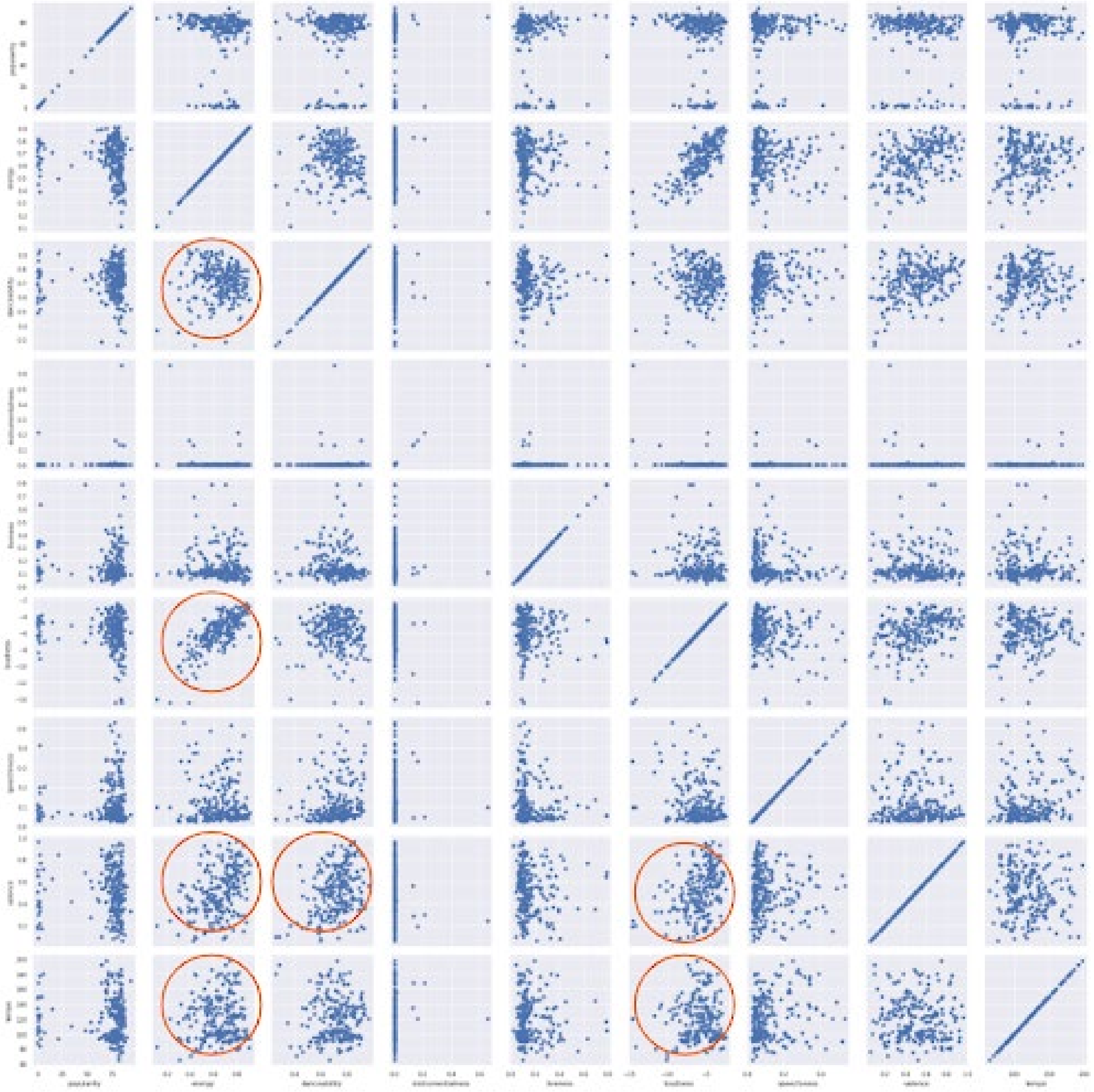
Most hit songs came out on Friday (230), regardless of rank, and this trend was consistent across the years. The ANOVA analysis showed there was no difference in Spotify rank associated with the day of the week the song was released.



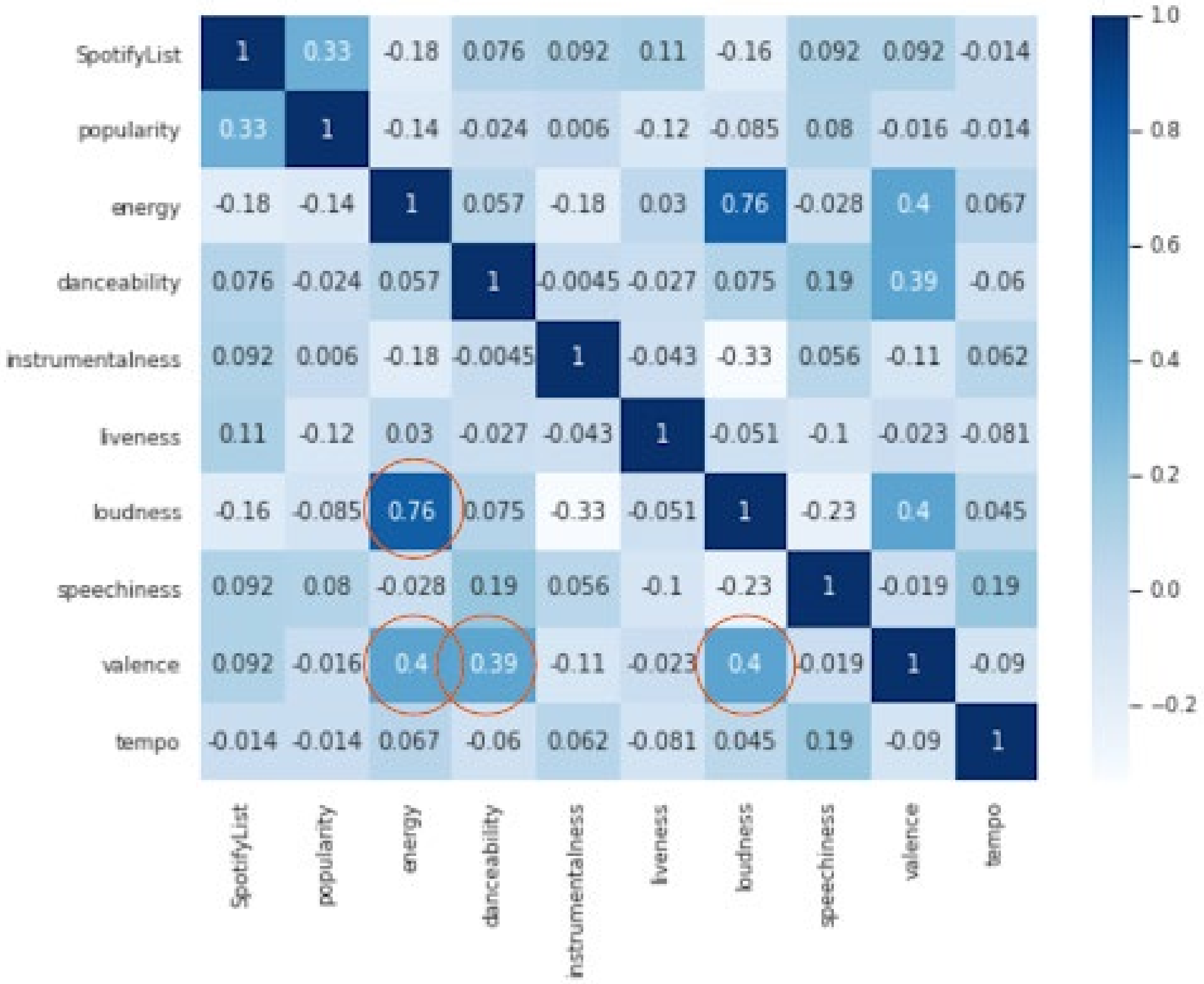


Song Attributes

We observed positive correlations among Energy & Danceability, Energy & Loudness, Energy & Tempo, Energy & Valence, Valence & Danceability, Valence & Loudness, Tempo & Loudness. In addition, a heatmap showed that the paired features with the highest correlations were Energy & Loudness, Energy & Valence, Valence & Danceability, and Valence & Loudness.

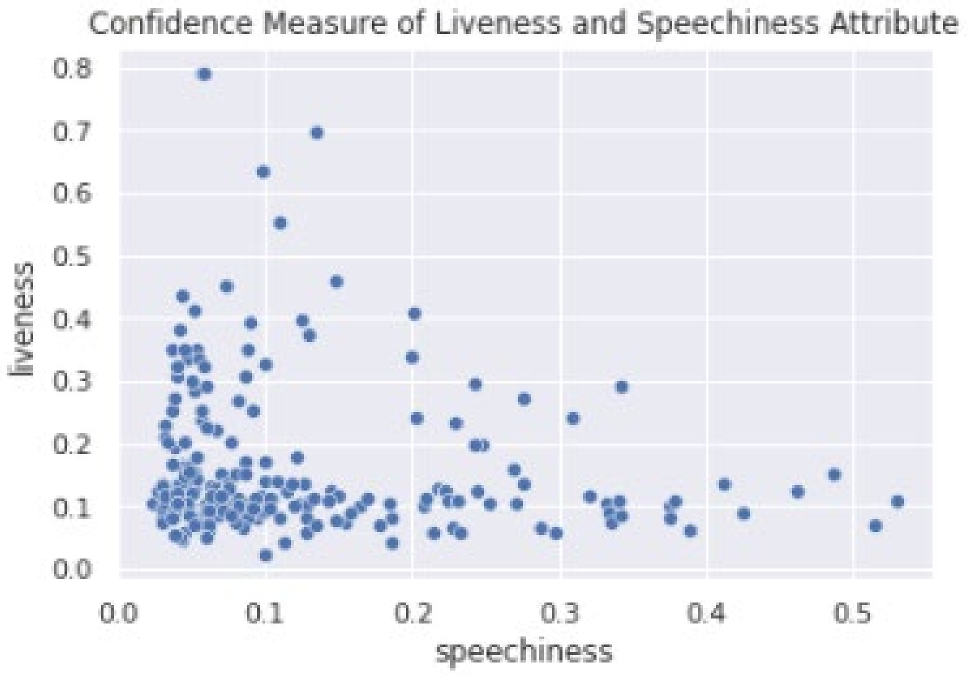


Song Attributes



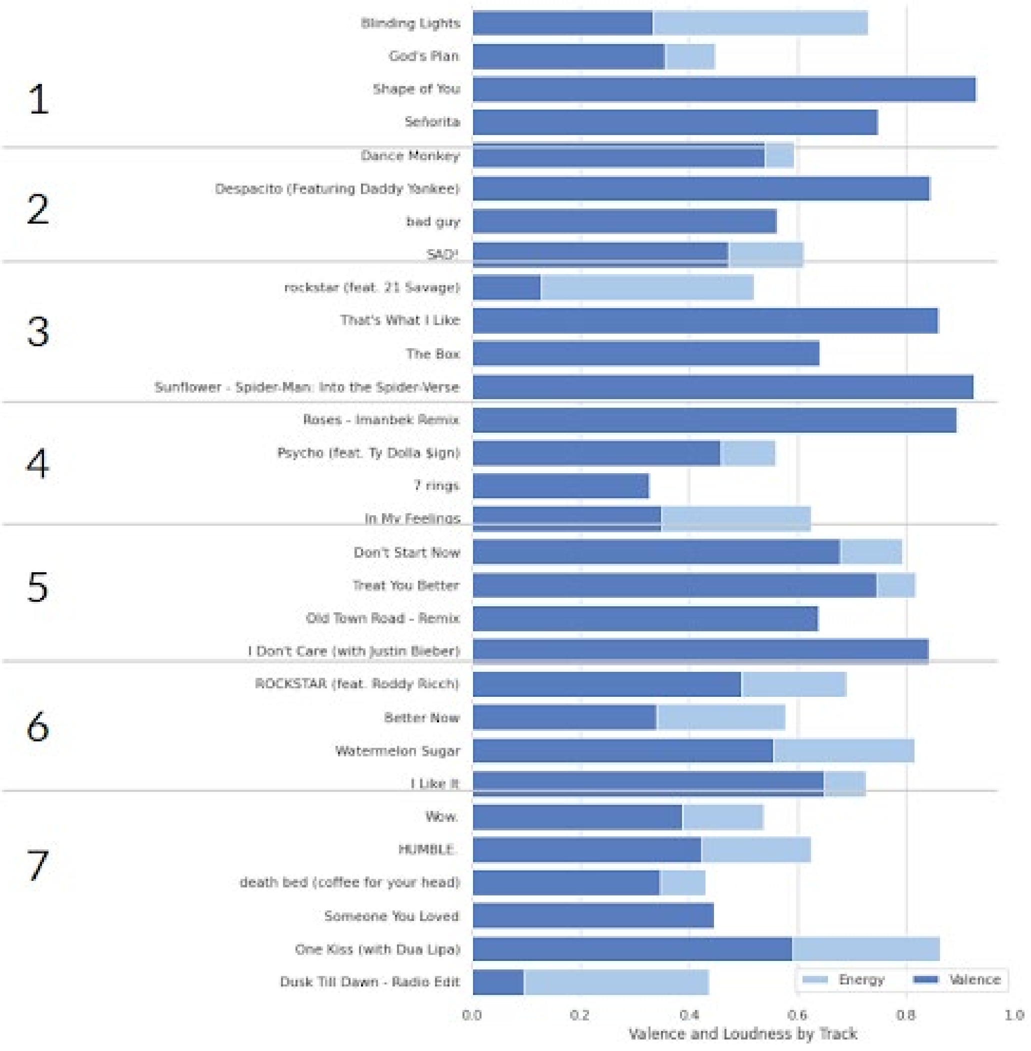
	SpotifyList	popularity	energy	danceability	instrumentalness	liveness	loudness	speechiness	valence	tempo
SpotifyList	1	0.33	-0.18	0.076	0.092	0.11	-0.16	0.092	0.092	-0.014
popularity	0.33	1	-0.14	-0.024	0.006	-0.12	-0.085	0.08	-0.016	-0.014
energy	-0.18	-0.14	1	0.057	-0.18	0.03	0.76	-0.028	0.4	0.067
danceability	0.076	-0.024	0.057	1	-0.0045	-0.027	0.075	0.19	0.39	-0.06
instrumentalness	0.092	0.006	-0.18	-0.0045	1	-0.043	-0.33	0.056	-0.11	0.062
liveness	0.11	-0.12	0.03	-0.027	-0.043	1	-0.051	-0.1	-0.023	-0.081
loudness	-0.16	-0.085	0.76	0.075	-0.33	-0.051	1	-0.23	0.4	0.045
speechiness	0.092	0.08	-0.028	0.19	0.056	-0.1	-0.23	1	-0.019	0.19
valence	0.092	-0.016	0.4	0.39	-0.11	-0.023	0.4	-0.019	1	-0.09
tempo	-0.014	-0.014	0.067	-0.06	0.062	-0.081	0.045	0.19	-0.09	1

We used a two-dimensional visualization to identify if there were any negative correlations between the song attributes. From the scatter plot, it was evident that there is a negative correlation between speechiness and liveness. The negative correlation present implies a seesaw effect will occur when liveness is at its highest point, speechiness at its lowest point.

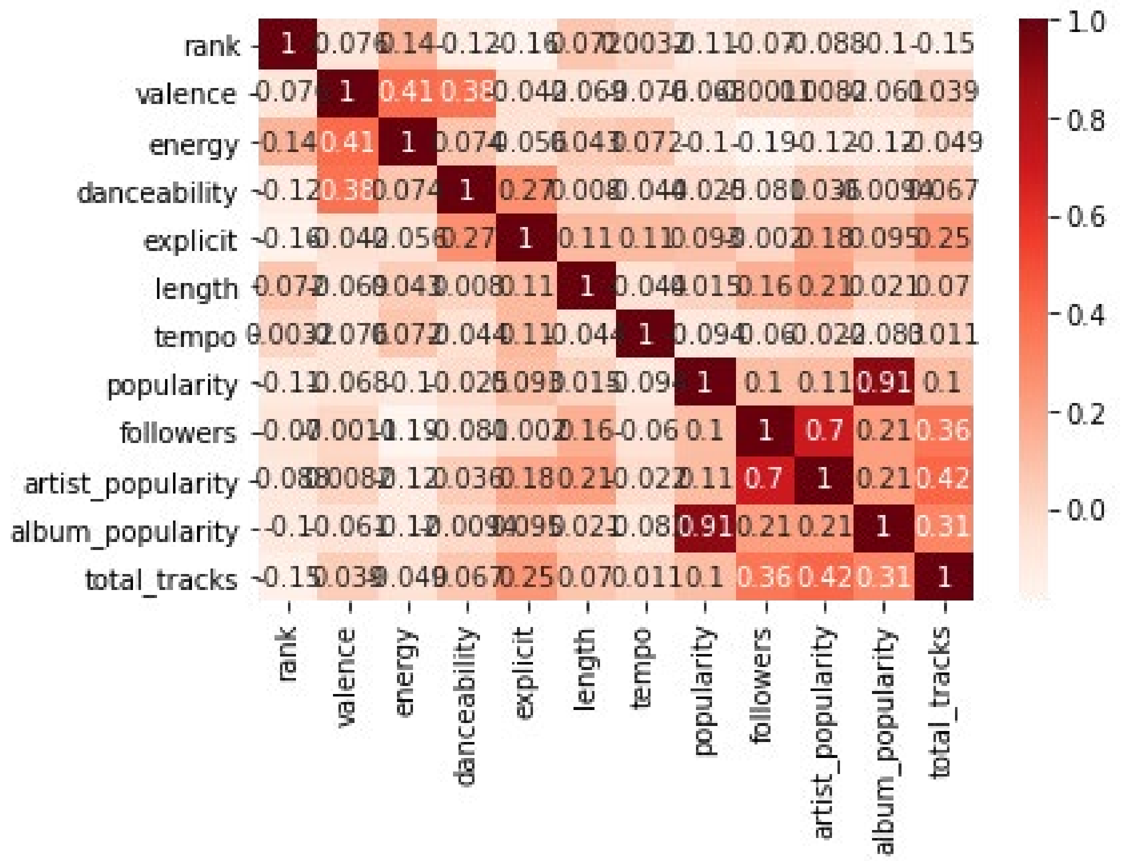


Song Attributes

After seeing that energy, valence, and loudness had the highest correlations, we looked at the top 7 songs and how energy and valence showed up. Loudness was challenging to plot because the values are negative and have a much larger scale. In 2020, two of the top 5 songs had high energy scores, but generally, energy starts to show up in lower-ranking songs at a higher rate.



Using a dataset filtered by rank, we analyzed how some of the audio features (valence, energy, danceability, and tempo) and popularity measures (followers, artist_popularity, album_popularity) correlated with each other. We first built a heatmap for the Top 50 tracks from each year. The paired features that have the highest correlation in this heatmap were: track popularity & album_popularity, artist_popularity & followers, and artist popularity & total tracks.

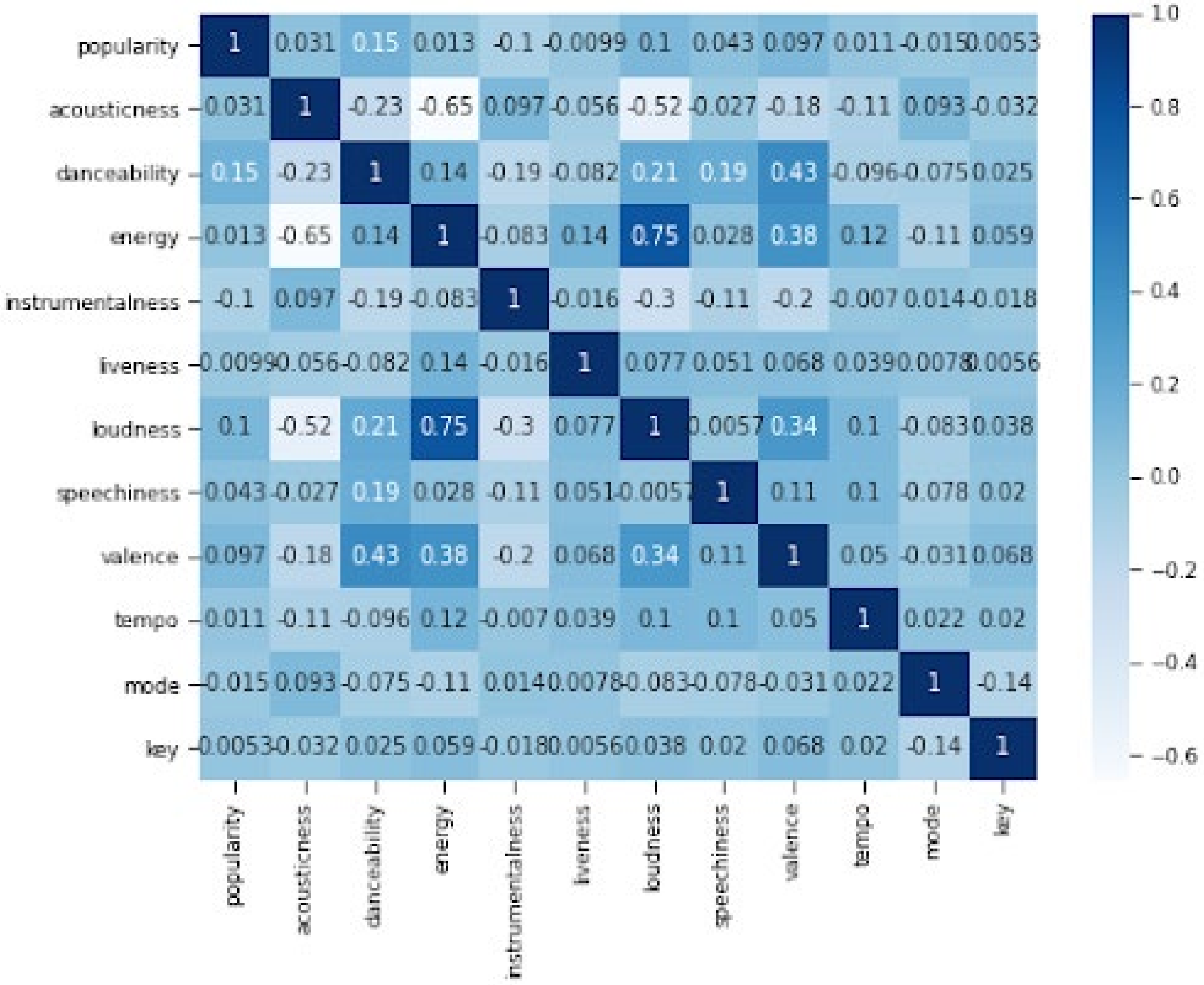


A second heatmap was created to filter the Top 10 tracks from each year. The paired features that have the highest correlation in this second heatmap are: track popularity & album_popularity, artist_popularity & followers, and artist popularity & total tracks. The correlation measures decreased when the data was filtered to Top 10 tracks, but there were no new correlations observed. The data shows that there is not much variability between the Top 50 and Top 10 tracks.

OBSERVATIONS FOR NON-HITS

Song Attributes

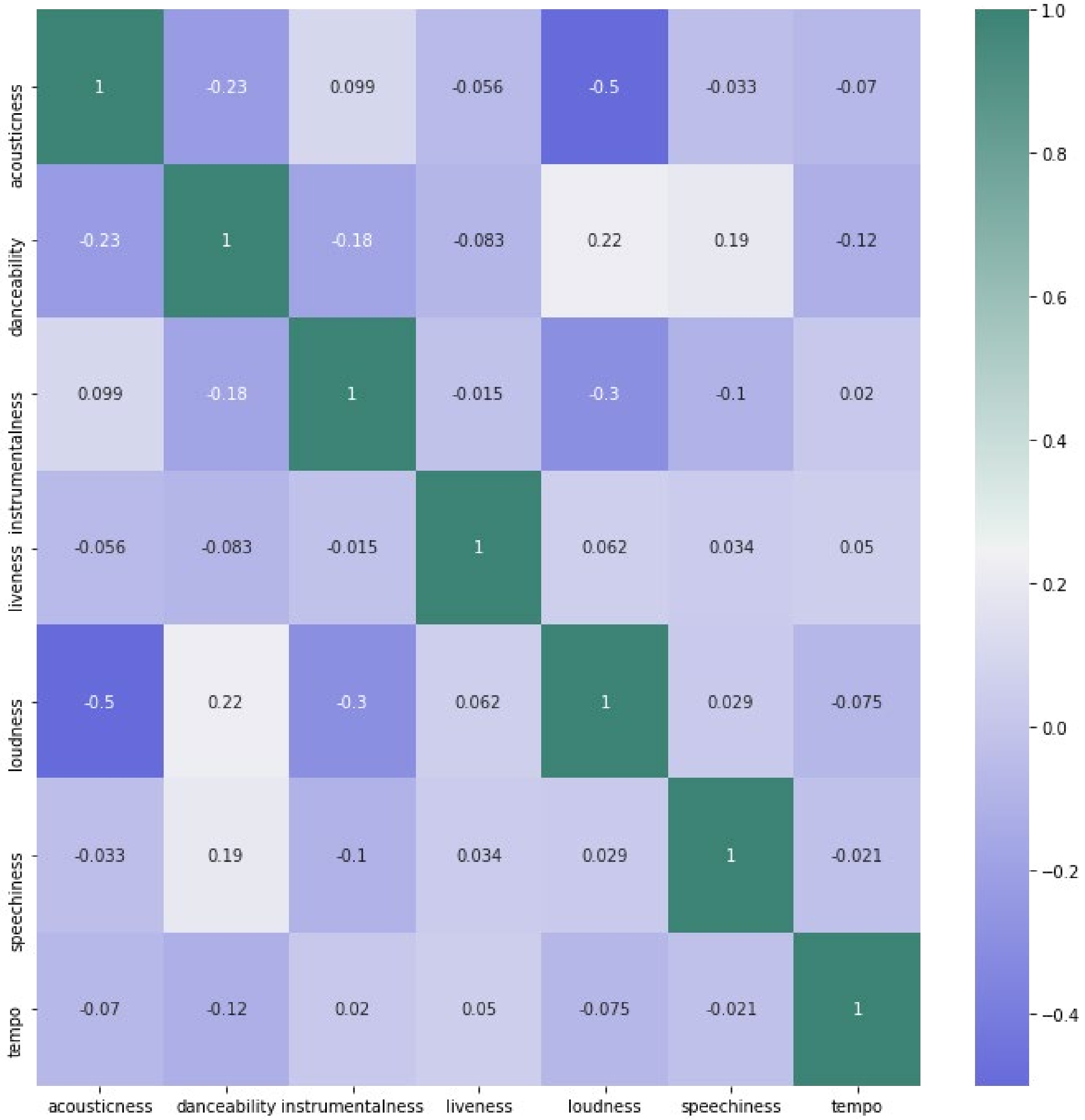
A heatmap of the song attributes for the non-hit tracks showed that the paired features with the highest correlations were Energy & Loudness, Energy & Valence, Valence & Danceability, and Valence & Loudness. This outcome was the same as the hit tracks.



OBSERVATIONS FOR COMBINED DATASET

Song Attributes

We found a moderate negative correlation between loudness and acousticness. Otherwise, there were no significant correlations between features.



2.C. STATISTICAL ANALYSIS & MODELING

OVERVIEW AND RATIONALE

We see an opportunity to provide record labels and artists with insights on the features of popular music that will assist in their creation process. Therefore, we sought to understand what combinations of song features (audio features, artist profile, label profile, etc.) determine song popularity on Spotify and Billboard to allow different stakeholders in the song production process to make informed, data-driven decisions.

We obtained data from the Spotify and Billboard API to create datasets of Spotify hit songs, Billboard hits songs, and non-hit songs between 2017-2020. Given a limited number of hit songs annually and no measure of popularity among non-hit songs, we decided to analyze our outcome as binary: hit song or not. For our intended customer, this information is valuable to determine what features are associated with a higher likelihood of a song being a hit song.

To make our results robust, we analyzed which features differentiated a hit from non-hit songs in the following ways:

1. Bivariate analysis (t-test and chi-square tests)
2. Within dataset prediction using multivariate logistic regression with training and testing sets: Spotify hit versus non-hit song, and Billboard hit versus non-hit song.
3. Out of sample prediction using multivariate logistic regression: Because songs on the Billboard hit list are determined by a more comprehensive set of criteria than Spotify hit songs (streaming frequency alone), we examined whether a model trained on Spotify hit versus non-hit songs, with previously identified features, would have predictive accuracy to distinguish Billboard hit versus non-hit songs.

Our exploratory data analysis focused on bivariate analyses (#1). First, we observed audio features that had noticeably different averages when comparing hits and non-hits. Next, we developed a hypothesis that hits could be identified from the unique combination of audio features. Finally, we performed t-tests to determine if the difference in audio features were statistically significant to support our claims.

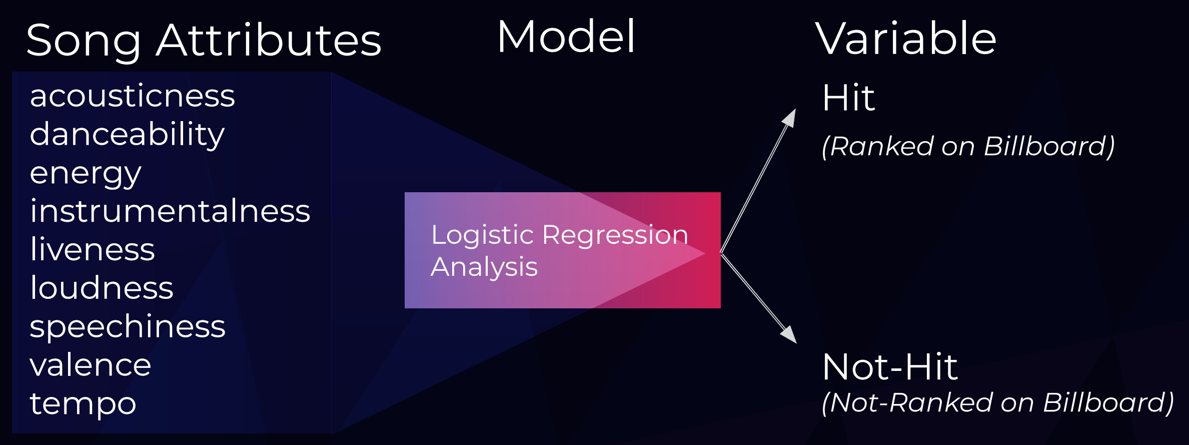
To further explore the relationship between audio features and whether a song is a hit, we decided to create classification models on subsets of our dataset. The subsets created around our models were: Spotify hits and Billboard hits. Furthermore, we were curious to train a model on Spotify data and test on Billboard data.

DATA PREPARATION AND ANALYSIS

We decided to use logistic regression to create a classification model. We met the necessary assumptions for fitting a logistic regression:

- our target variable is binary
- the observations are independent
- there is an absence of multicollinearity
- there are no outliers
- logistic regression does not demand too many computational resources

The first step in preparing our data was normalizing the numerical features. We used a min-max scalar method to have all our numerical features on a 0 to 1 scale. The next step was to create the dataframes for the training and testing. The first dataframe included non-hits and hits on Spotify. The second had non-hits and hits found on the billboard list. Next, we made sure there was no missing data or duplicates in either of our dataframes. Next, we created our target variable, hits, using the which_list variable. Finally, we categorized songs on a Spotify list, billboard list, or both as a hit.



DATA PREPARATION AND ANALYSIS

We used a bivariate analysis to select our features to include in the final logistic regression model. However, because we had imbalanced datasets of hit vs. non-hits and lack of normality, we had to use specialized tests to compare measures of centralized tendency. Specifically, since the feature had to be significant in both Welch and Yuen's t-test, we realize that this may be overly strict, but we wanted to have a parsimonious model that could be digested and actionable by our intended client. Therefore, we decided to include the features in our models: explicit, acousticness, danceability, instrumentalness, liveness, loudness, speechiness, and tempo.

Welch t-test results

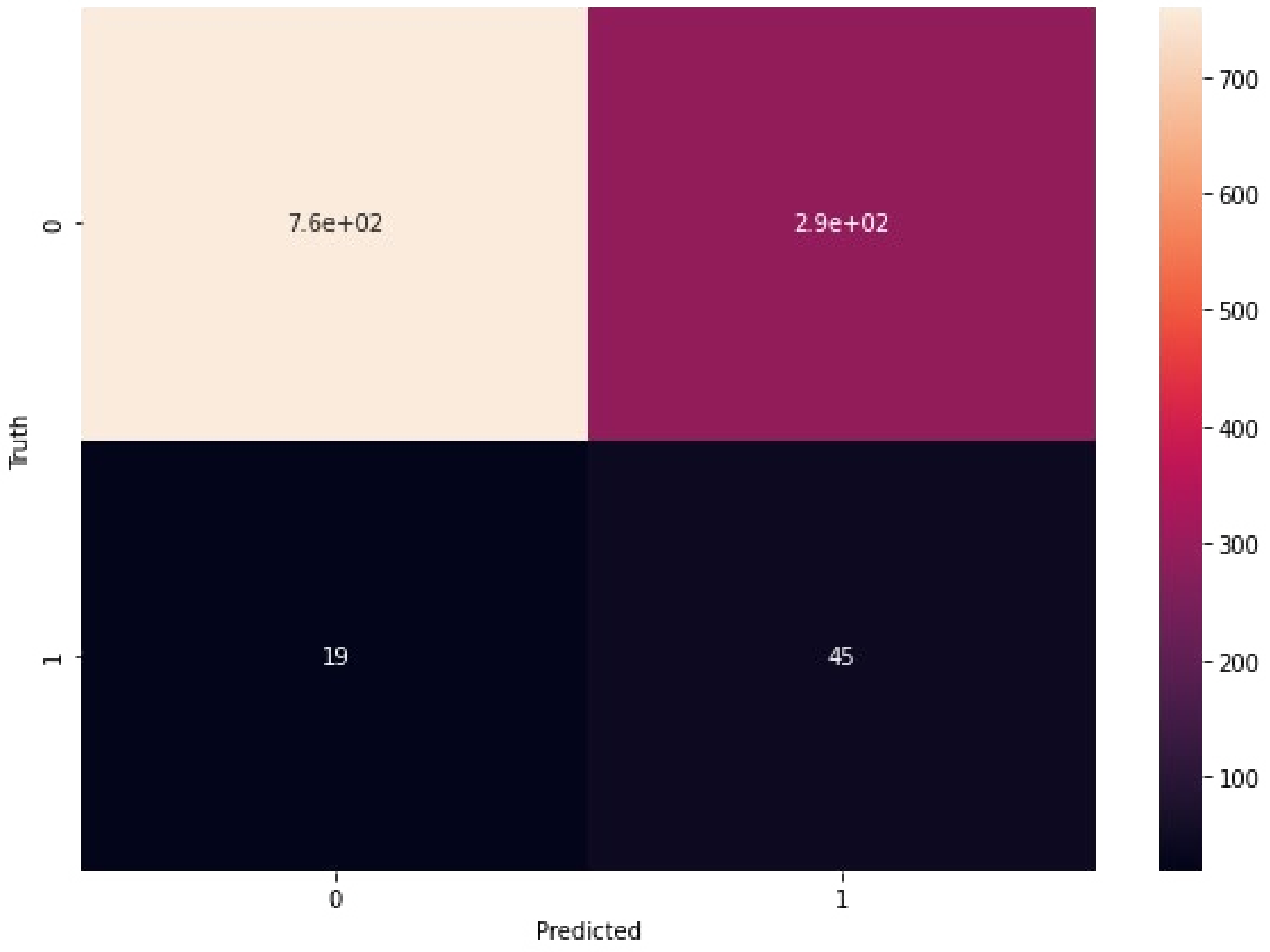
Variable	T-statistic	P-value	95% CI	Cohen's D
Liveness	-3.03	0.003	-0.03, -0.01	0.14
Loudness	-7.34	<0.001	0.1, 0.11	0.80
Speechiness	6.24	<0.001	0.07, 0.11	0.67

We relied on the libraries and methods in the sci-kit learn package to implement our logistic regression. To split our data, we used 80% of our data to train the model and 20% to test. We then specified a logistic regression model and a class weight to account for the imbalance data. For robustness, we also tested our within-dataset logistic regression model with a balanced dataset by undersampling the non-hit songs, which were more numerous than hit songs. We also conducted an out of sample prediction using a model training on Spotify data and tested on Billboard data.

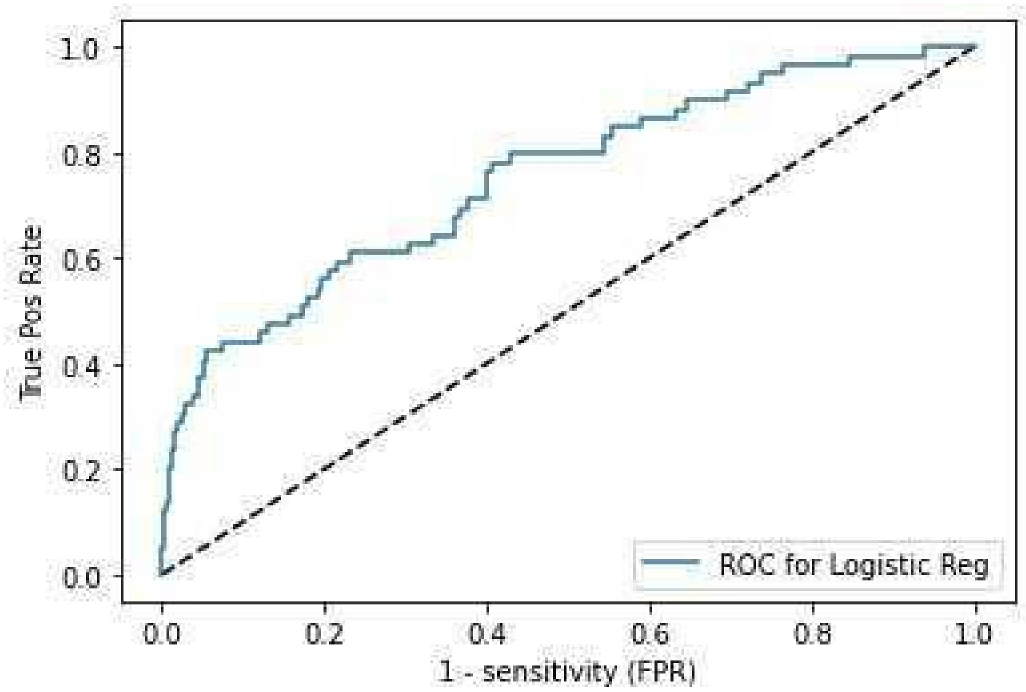
Prediction	ROC AUC
Within Spotify	0.72
Within Spotify undersampling non-hits	0.66
With Billboard	0.85
Out of sample	0.73

Our models had high accuracy in distinguishing hits vs non-hits

Spotify hit versus non-hit classification (imbalanced data)



Out of sample Receiver Operating Curve (Spotify)



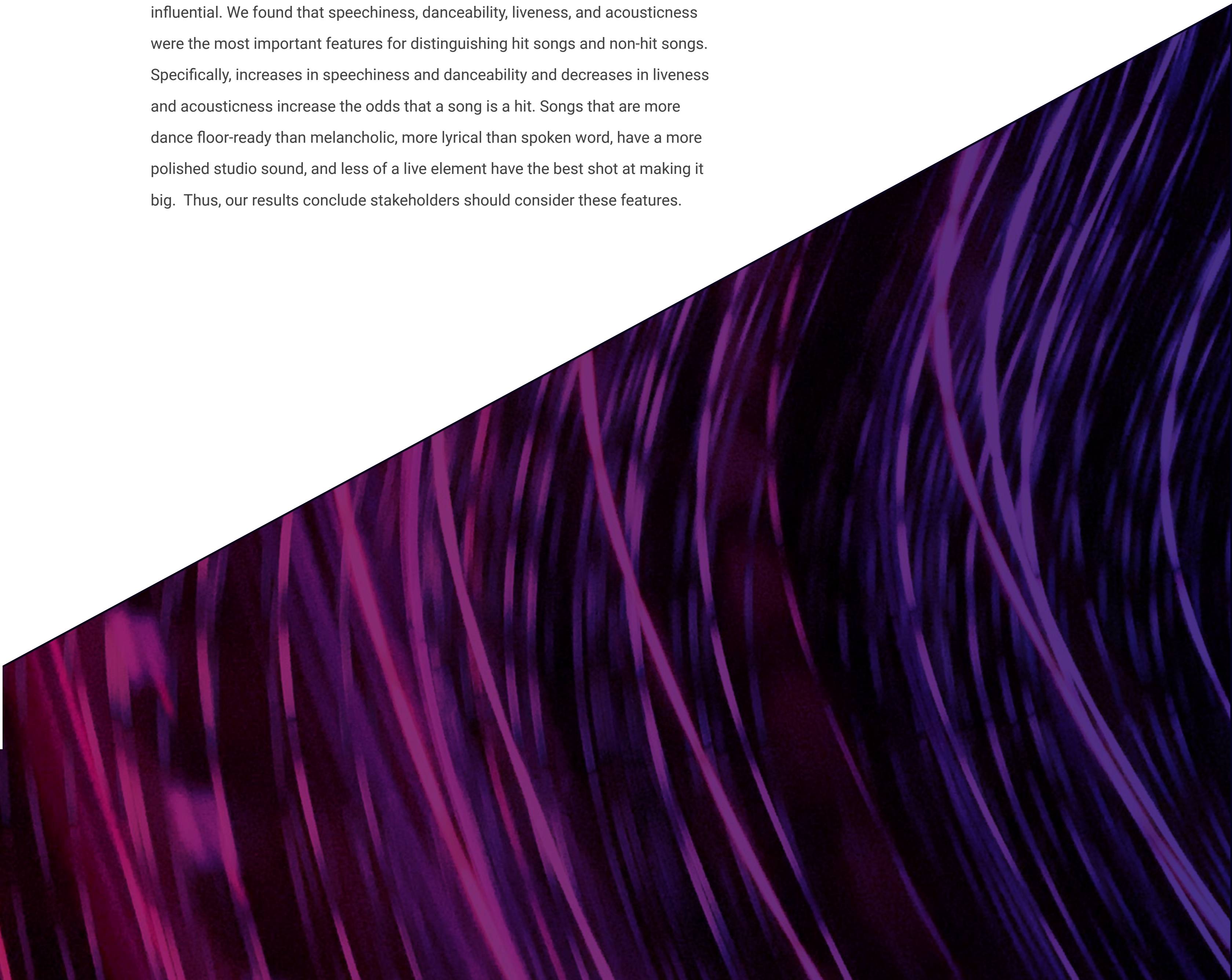
Most Important Features

Spotify Model *(odds ratio)*	**Billboard Model** *(odds ratio)*	**Out of sample** *(odds ratio)*	**T-tests** *(Cohen's d)*
Speechiness	Danceability	Speechiness	Loudness
Danceability	Speechiness	Danceability	Speechiness
Liveness	Tempo	Liveness	Tempo
Acousticness	Liveness	Acousticness	Energy

Conclusions

Using acoustic features obtained from Spotify, we found that models including loudness, speechiness, danceability, explicit, instrumentalness, acousticness, liveness, and tempo had high accuracy (as measured by precision, recall, F1 score, and AUC on ROC plots) for classification of hit versus non-hit songs on both Spotify and Billboard top songs. Furthermore, a model trained on Spotify hits and non-hit using these acoustic features had high accuracy in distinguishing Billboard hits vs. non-hits. Thus, we conclude that acoustic features are a valuable way to determine hit songs from non-hit songs.

To make these findings more tangible, we determined which features were most influential. We found that speechiness, danceability, liveness, and acousticness were the most important features for distinguishing hit songs and non-hit songs. Specifically, increases in speechiness and danceability and decreases in liveness and acousticness increase the odds that a song is a hit. Songs that are more dance floor-ready than melancholic, more lyrical than spoken word, have a more polished studio sound, and less of a live element have the best shot at making it big. Thus, our results conclude stakeholders should consider these features.



03

DASHBOARD DESCRIPTION

3.A. USE CASES

3.B. TECHNOLOGIES USED

3.C. VISUALIZATION FEATURES

3.D. DATA ENGINEERING

3.A. USE CASES

Our goal was to allow the musical stakeholders to interact with the data. Therefore, the main questions we wanted to answer for them were:

- Which attributes make a song a hit?
- What attributes have worked in the past?
- Which attributes differ between hit songs and non-hit songs?

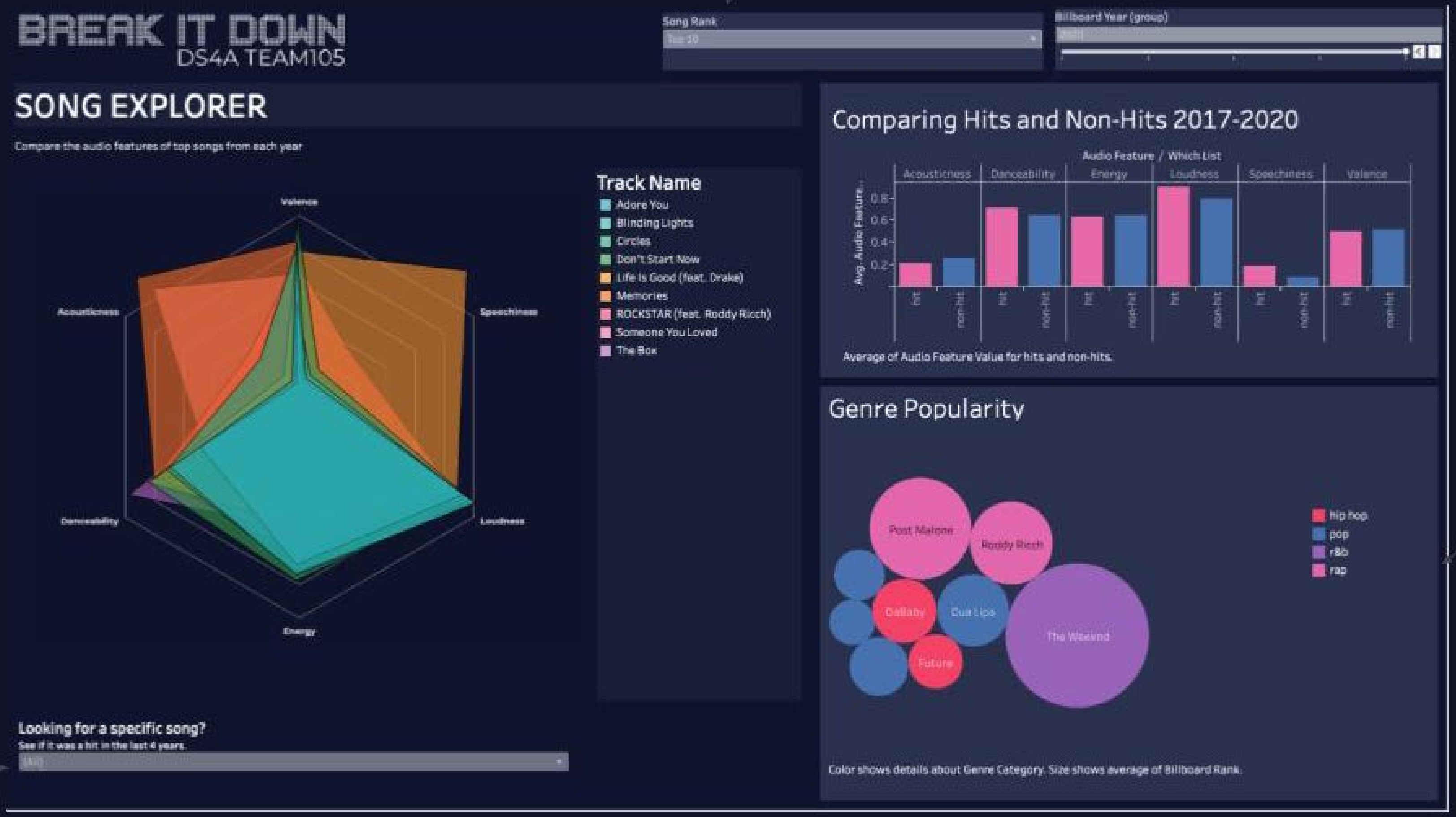
3.B. TECHNOLOGIES USED

We built our interactive dashboard in Tableau

3.C. VISUALIZATION FEATURES

After reviewing the visualizations from our EDA, we identified the three views that would provide the most value and fun for our intended audience.

Dashboard



SONG EXPLORER

Compare the audio features of top songs from each year

Looking for a specific song?

See if it was a hit in the last 4 years.

Comparing Hits and Non-Hits 2017-2020

Average of Audio Feature Value for hits and non-hits.

Genre Popularity

Color shows details about Genre Category. Size shows average of Billboard Rank.

Filter by Song Rank

Filter by Year

Radar plot with audio attributes, filter by rank and year

Select individual songs to inspect

Changes in genre

DS4A | TEAM 105

SONG EXPLORER

As our most ambitious view to build, we chose radar plots to display the audio attributes for each song. The selected attributes included the crucial attributes determined by our regression analysis and factored in our understanding of the meaning behind the features. For instance, we did not include "liveness" even though it was an essential attribute in our regression analysis simply because the meaning behind it was whether or not the artist performed a song in front of an audience. The majority of hit songs are recorded in studios. We also omitted instrumentalness because the differences in the values were too small to see in the graph.

We tied this chart to the macro filters on the dashboard to sort by rank (Top 10, 25, 50, 100) and by year. There is a key showing which songs are displayed and the corresponding color on the radar plot. There's an option for a person to select a specific song from the drop-down menu to view it, regardless of the macro filters.

COMPARING HITS AND NON-HITS

A bar chart was the most straightforward and most elegant view we found for comparing hits and non-hits side by side. We chose to feature the same attributes selected in our "song explorer" radar plots based on the regression model.

GENRE POPULARITY

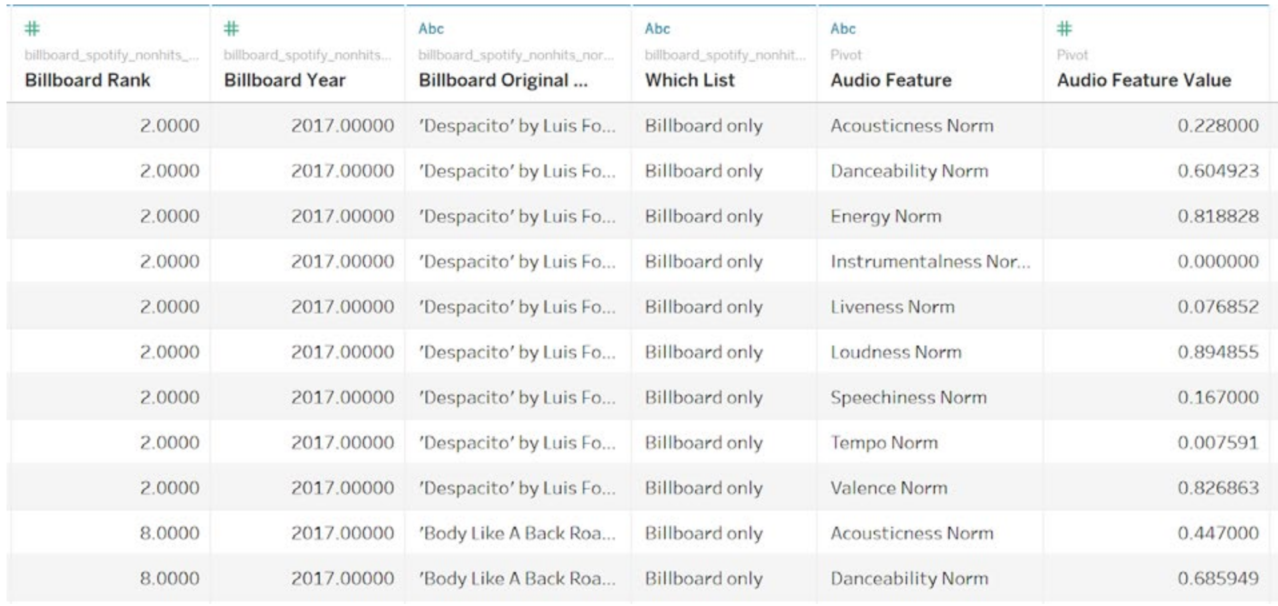
As we reviewed our data in Python, we thought it was interesting to see how the popular genres changed each year. For example, we were surprised to see that the top songs weren't always pop songs, but that instead, there was a good mix of genres that changed each year. A stacked bubble chart gave us a fun way to display that kind of information. The macro filters can also sort this chart for rank and year.

3.D. DATA ENGINEERING

Working from our final schema, we still found some changes we needed to make to our dataset to build the visualizations and give a complete picture to project stakeholders.

The main change was normalizing the values of our song attributes - we wanted them all as values between 0 and 1 so that someone could visually distinguish the differences between attributes for each song. To get all the audio features on a 0 to 1 scale, we applied a min-max normalization to acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, valence, and tempo.

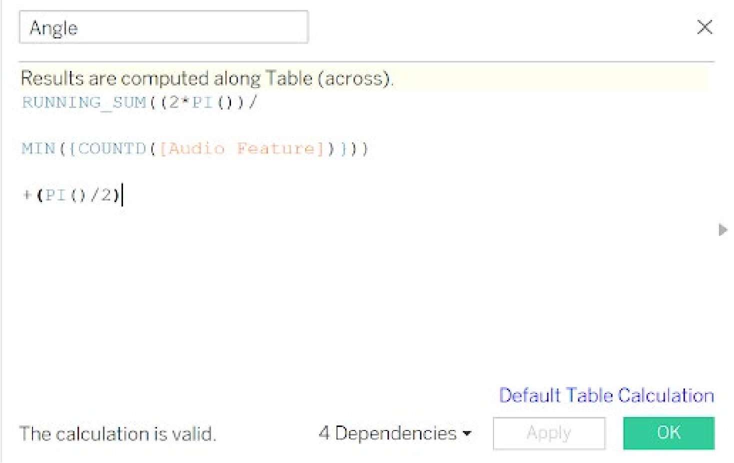
After normalizing the data, we had to create a pivot inside of Tableau (see image below) so that the attributes were all in one column (Audio Feature) and their values in another (Audio Feature Value). We used the pivoted column for both the "Song Explorer" and "Comparing Hits and Non-Hits" plots. Another calculated field we needed was for the rankings. We created groups for the Top 10, 25, 50, and 100 to make easier comparisons and groupings in the visualization.



# billboard_spotify_nonhits_... Billboard Rank	# billboard_spotify_nonhits... Billboard Year	Abc billboard_spotify_nonhits_nor... Billboard Original ...	Abc billboard_spotify_nonhit... Which List	Abc Pivot Audio Feature	# Pivot Audio Feature Value
2.0000	2017.00000	'Despacito' by Luis Fo...	Billboard only	Acousticness Norm	0.228000
2.0000	2017.00000	'Despacito' by Luis Fo...	Billboard only	Danceability Norm	0.604923
2.0000	2017.00000	'Despacito' by Luis Fo...	Billboard only	Energy Norm	0.818828
2.0000	2017.00000	'Despacito' by Luis Fo...	Billboard only	Instrumentalness Nor...	0.000000
2.0000	2017.00000	'Despacito' by Luis Fo...	Billboard only	Liveness Norm	0.076852
2.0000	2017.00000	'Despacito' by Luis Fo...	Billboard only	Loudness Norm	0.894855
2.0000	2017.00000	'Despacito' by Luis Fo...	Billboard only	Speechiness Norm	0.167000
2.0000	2017.00000	'Despacito' by Luis Fo...	Billboard only	Tempo Norm	0.007591
2.0000	2017.00000	'Despacito' by Luis Fo...	Billboard only	Valence Norm	0.826863
8.0000	2017.00000	'Body Like A Back Roa...	Billboard only	Acousticness Norm	0.447000
8.0000	2017.00000	'Body Like A Back Roa...	Billboard only	Danceability Norm	0.685949

RADAR PLOT

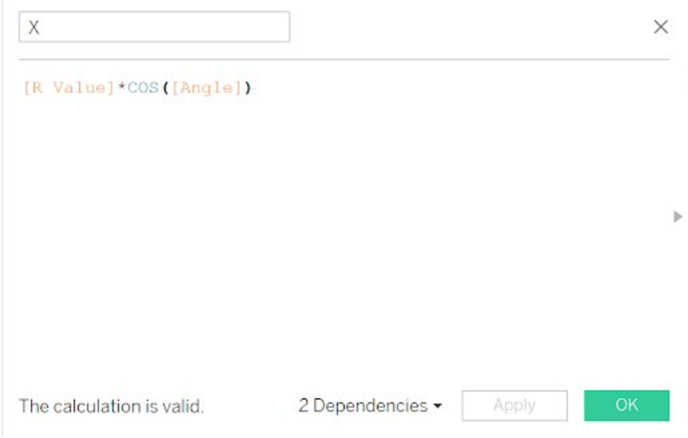
To create the radar plot, we had to create four different calculated fields to account for the fact that radar plots are circular and use polar coordinates, but Tableau works in Cartesian coordinates[5]. The first calculation was for the angle of the radar chart where RUNNING_SUM allows us to travel around the increments of the circle, 2*PI() is the revolution of a circle, and MIN({COUNTD([Audio Feature])}) counts the number of dimensions within Audio Feature and divides the circle by the distinct number of features. The second calculation is for the r-value, the distance between the origin and the data point. For our project, we did AVG([Audio Feature Value]). Using the angle and r-value, we then created two separate calculations, one for the X and Y positions on the plot.



Angle

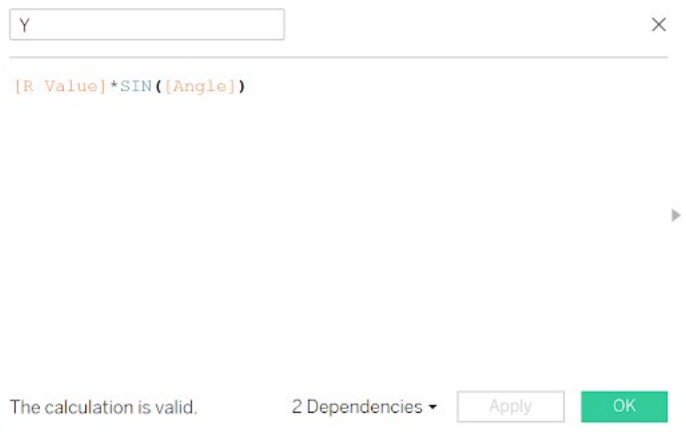
Results are computed along Table (across).

```
RUNNING_SUM((2*PI())/

MIN({COUNTD([Audio Feature])}))

+(PI()/2)
```



X

```
[R Value]*COS([Angle])
```

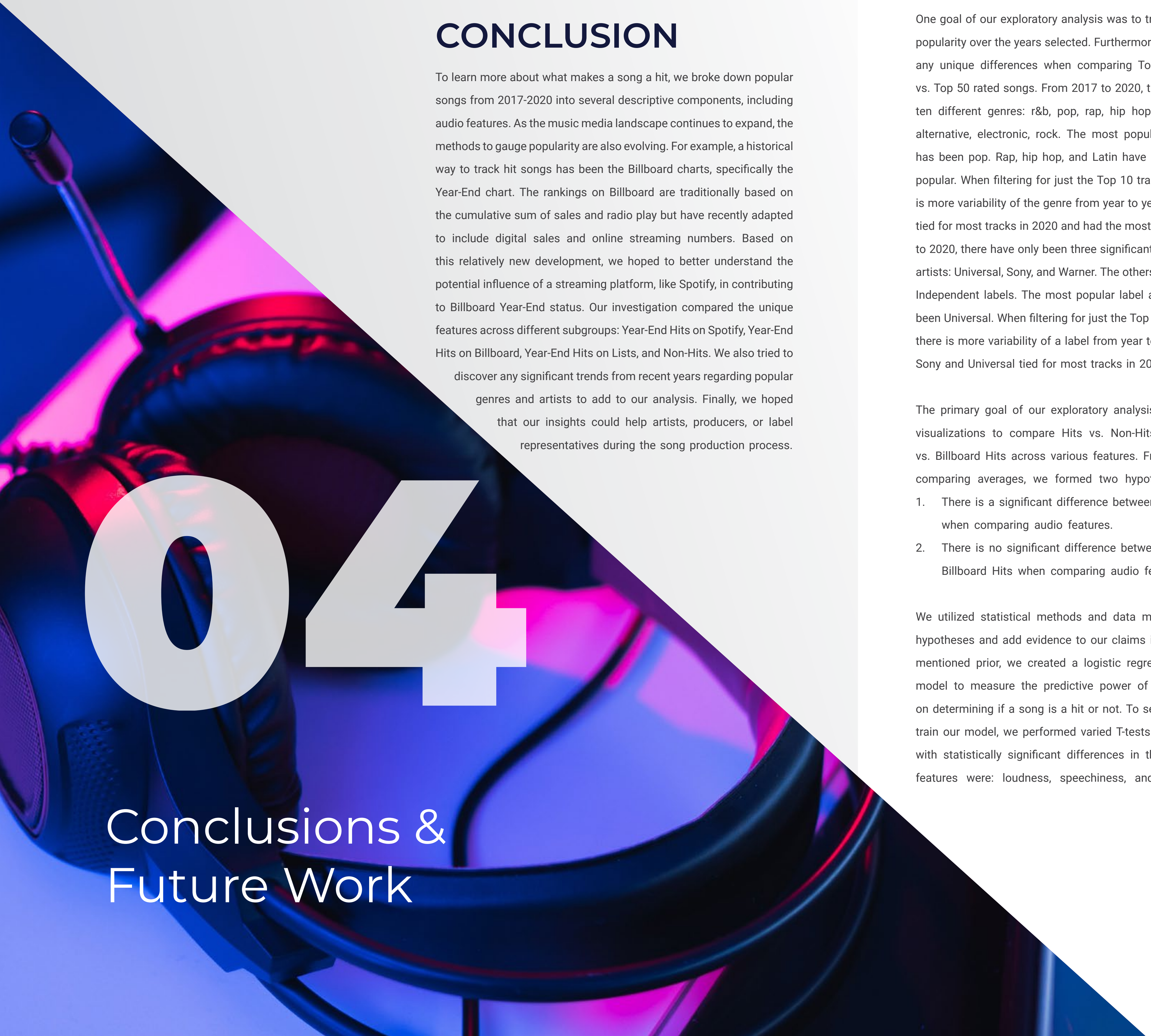


Y

```
[R Value]*SIN([Angle])
```

GENRE STACKED BUBBLE CHART

Since a top rank is represented by a lower numerical value (1,2, 3...) and a lower rank is a higher numerical value (100, 99, 98), the bubble chart was initially showing bigger circles for lower ranking songs. To correct this, we created a calculated field where we divided 1 by rank.



04

Conclusions & Future Work

CONCLUSION

To learn more about what makes a song a hit, we broke down popular songs from 2017-2020 into several descriptive components, including audio features. As the music media landscape continues to expand, the methods to gauge popularity are also evolving. For example, a historical way to track hit songs has been the Billboard charts, specifically the Year-End chart. The rankings on Billboard are traditionally based on the cumulative sum of sales and radio play but have recently adapted to include digital sales and online streaming numbers. Based on this relatively new development, we hoped to better understand the potential influence of a streaming platform, like Spotify, in contributing to Billboard Year-End status. Our investigation compared the unique features across different subgroups: Year-End Hits on Spotify, Year-End Hits on Billboard, Year-End Hits on Lists, and Non-Hits. We also tried to discover any significant trends from recent years regarding popular genres and artists to add to our analysis. Finally, we hoped that our insights could help artists, producers, or label representatives during the song production process.

One goal of our exploratory analysis was to track genre and label popularity over the years selected. Furthermore, we wanted to test any unique differences when comparing Top 10 ranked songs vs. Top 50 rated songs. From 2017 to 2020, there have only been ten different genres: r&b, pop, rap, hip hop, Latin, soul, k-pop, alternative, electronic, rock. The most popular across all years has been pop. Rap, hip hop, and Latin have been the next most popular. When filtering for just the Top 10 tracks each year, there is more variability of the genre from year to year. For example, rap tied for most tracks in 2020 and had the most in 2018. From 2017 to 2020, there have only been three significant labels representing artists: Universal, Sony, and Warner. The others include Disney and Independent labels. The most popular label across all years has been Universal. When filtering for just the Top 10 tracks each year, there is more variability of a label from year to year. For example, Sony and Universal tied for most tracks in 2017 and 2020.

The primary goal of our exploratory analysis was to use quick visualizations to compare Hits vs. Non-Hits, and Spotify Hits vs. Billboard Hits across various features. From our analysis of comparing averages, we formed two hypotheses:

1. There is a significant difference between Hits and Non-Hits when comparing audio features.
2. There is no significant difference between Spotify Hits and Billboard Hits when comparing audio features.

We utilized statistical methods and data modeling to test our hypotheses and add evidence to our claims in our next step. As mentioned prior, we created a logistic regression classification model to measure the predictive power of our audio features on determining if a song is a hit or not. To select the features to train our model, we performed varied T-tests to identify features with statistically significant differences in the averages. These features were: loudness, speechiness, and danceability. The accuracy, precision, and recall for our model were substantial. This analysis allowed us to conclude that based on Spotify's few influential audio features, we can identify songs as Non-Hits or Hits. Furthermore, Spotify and Billboard share the same significant audio features. In terms of our original business objective, these results allow us to give those involved in the song production process elements of music to prioritize to align with the trends of popular or successful songs.

As part of our analysis, our dashboard allows users to explore further how audio feature combinations vary within popular songs by using our Song Explorer. A user can select specific songs to compare using the layering function of the visualization to provide a more precise reflection of differences. We also offer the opportunity to observe genre trends which is the area with the most variability over the years.

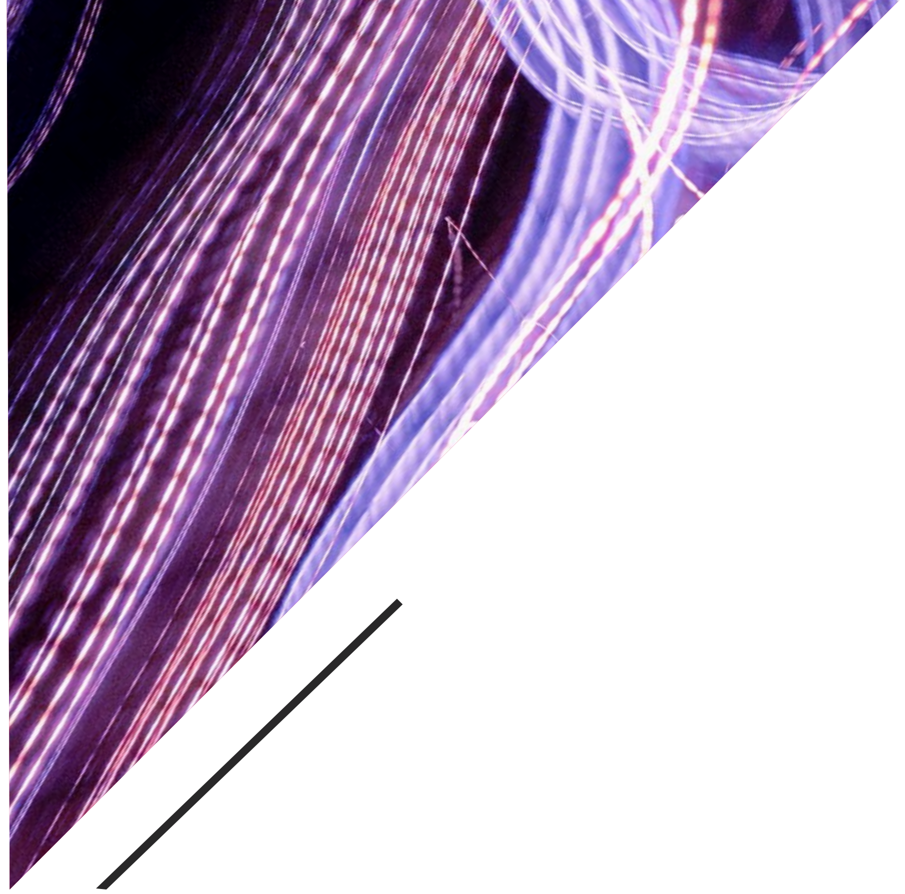
As we wrap up the first phase of our capstone project, we have reflected on potential improvements or developments for the subsequent phases. To start, we would like to gather more data to provide more context around these specific songs. There are a lot of factors that contribute to the success of a song beyond the music itself. It would be helpful to our analysis to look at streaming, financial, and marketing metrics. In general, we care to understand better social factors, like demographics of listeners across different platforms and social media impacts. In terms of social media, we would like to explore songs that trend on Tik Tok. As far as our dashboard, we would like to implement the ability to search for any song by having the Spotify API integrated to allow users to compare any songs of their choice if they have a particular interest or need. Finally, we would like to explore other classification models beyond logistic regression.

REFERENCES

1. Hissong, Samantha. “More Songs Are Going Platinum Than Ever Before.” Rolling Stone, Rolling Stone, www.rollingstone.com/pro/news/music-streaming-riaa-gold-platinum-songs-2020-1113311/.
2. “Billboard Year-End.” Academic Dictionaries and Encyclopedias, en-academic.com/dic.nsf/enwiki/2025281.
3. Music Gateway. “Music Labels: What Are They and a Review of the Top Record Labels.” Music Gateway, 22 Aug. 2019, www.musicgateway.com/blog/how-to/music-labels-top-record-labels.
4. “Spotify Web API Reference.” Spotify for Developers, developer.spotify.com/documentation/web-api/reference/#category-tracks.
5. “The Data School - a Simple Way to Make a Radar Chart.” The Data School UK RSS, www.thedataschool.co.uk/ellen-blackburn/a-simple-way-to-make-a-radar-chart.
6. Global Music Market Overview: https://gmr2021.ifpi.org/report

THANK YOU

1. Our amazing mentors Lena Evans and Kevin O’Sullivan who generously gave us time, feedback and shared their personal stories to give us inspiration and motivation when we needed it.
2. We had wonderful TAs Sia Seko and Victoria Morgan who graciously met with us weekly, delivered kick-ass lectures and were available to answer even the smallest question when we needed it.



FINAL REPORT

DS4A TEAM 105