Temporal Trends in Music Popularity – A Quantitative analysis of Spotify API data

Preprint · December 2018

DOI: 10.13140/RG.2.2.11551.71843

4 authors, including:

Tanner O'Rourke
University of Colorado Boulder
2 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Temporal Trends in Music Popularity

A quantitative analysis of Spotify API data

Matthew Menten
Computer Science- CSCI 4502
SID: 106634028
CU-Boulder
Matthew.Menten@colorado.edu

Kieren Ng
Computer Science- CSCI 4502
SID: 103692884
CU-Boulder
king6693@colorado.edu

Braden Holmes
Computer Science- CSCI 4502
SID: 106080729
CU-Boulder
brho9944@colorado.edu

Tanner O'Rourke
Computer Science- CSCI 4502
SID: 106523745
CU Boulder
Boulder, CO, US
taor0213@colorado.edu

Figure 1 - Spotify Logo

Abstract

Music is the most accessible and widely popular form of artistic expression, affecting our the lives of millions everyday. While music dates back as far as the modern human, it has also had mass effect on society in the last 10 years - making it vital for us to understand its trends and their significance going into the future. Many statistical analyses have been conducted on musical trends, such as Shane Snow's analysis of key verbiage and content in songs since 1965. These works generally focus on the qualitative attributes such as verbiage of musical pieces. While powerful, this doesn't provide a analysis of songs in their purest form - their generalized attributes.

Our approach aims to analyze *the way songs are*, not the "certain qualities a song holds". We hope to utilize this different approach to not just define historical music trends, but identify trends to predict the how music will be in the coming years.

We believe that our key results could have great value to the music industry by giving an early glimpse of comprehensive music trends. This would allow artists, producers, and writers alike be able to know their audience on a greater level, and create songs that are predictably successful. Further, artists could predict what will be popular in the future and gain an advantage over others in the market. Additionally, analyzing the changes in music can provide greater insight into the human population as a whole. For example, if a large demographic majority are listening to happier music than people twenty years ago, we might hypothesize that people are relatively happier than they used to be. Throughout this project, we were able to classify the danceability of a song with greater than 70% accuracy, found that acoustic songs are becoming less

prevalent while higher energy songs are on the rise and found that the distribution of keys followed the notes of the C-major scale.

Introduction

Motivation

Music is the most culturally and emotionally ingrained art form in our current cultural climate. So much that deciphering a change in human demeanors can imply a change in the human consciousness as a whole. Developments and trends in modern music as they compare to the past can thus explain current and future societal and cultural trends.

Prior Work, Limitations

There have been numerous studies done on the attributes and specific qualities of music. Some of these include:

- Analyzing 50 years of Pop Music, by Shane Snow [2]
    - Uses the Billboard top charts from multiple decades.
    - Uses graphical displays to analyze music by decade based on key verbiage, content, and demographics.
- How Music has evolved in the last 70 years, by The Echo Nest [1]
    - Uses references to discuss major trends in music as it pertains to the past and present.
- Songs over the Decades - a Test Mining Analysis, by 'columnist' [3]
    - A graphical analysis of major musical themes such as filler words, general themes, and specific verbiage.
- Data Mining Applied to Music Style Classification [4]
    - Aims to build a music classifier based on

Many previous studies, papers, and research, as well as those shown above, aiming to study musical trend analysis, analyze more specific musical qualities. They primarily focus on qualitative attributes and specific qualities songs hold such as verbiage,, or broader topics such as genre and demographics. For example, its typical to make the assumption that the popularity of certain genres like rock-n-roll has become less mainstream and others such as "EDM" and "house" music have taken other genres place in recent years, yet it becomes much more difficult to know how the base qualities of a song have trended.

Our approach to the problem

In our project, we aim to measure musical trends in a more quantifiable way. Such as the fact that songs over time have become louder and more energetic.[1] Through the Spotify API, we will be able to access pre-calculated values for these more generalized attributes of music. By using a sufficiently large dataset distributed over many years, we will be able to collect and analyze this data seamlessly. Through this approach, we hope to gain a more in-depth understanding of how music has changed over time and in what ways.

Individual Contributions

Matthew Menten: API work, database creation, initial data cleaning, logistic regression, knn classification

Kieren Ng: SQL queries, API work, data cleaning, apriori algorithm implementation

Braden Holmes: Temporal plotting and 3d histograms

Tanner O'Rourke:API work, K-mean clustering, correlations over time

Organization of our Report

Our report begins by describing our methodology, meaning how we went about choosing a topic, retrieving our data set, and beginning analysis. Then, in the evaluation section we describe much of our findings and how we found it. In the discussion section, we talk about what went wrong and what we learned. In addition, we mention some future work that could be done. In the conclusion, we summarize our results and the project as a whole.

Methodology

Problem Formulation

Our projects' goal is to analyze quantifiable trends in music over the past many years. We hoped that through our analysis we could identify these quantitative trends in past music to predict both quantitative and qualitative qualities of music to the future. To do this, we needed to find a large dataset of songs that are evenly distributed over many years, genres, and qualities. Only then could we ensure that we are correctly analyzing past musical qualities in an evenly distributed fashion. As well, the dataset must be able to be used in-line with the Spotify API.

The Dataset

Our first task was to obtain a large set of songs. After a bit of research, we decided that the Million Songs Dataset[6] would best suit our needs for this project since we could download a text file of track names and artists which we could then use to query the Spotify API (as opposed to downloading the ~300 Gb of song files). Our main challenge here was to select the right amount and range of songs. Data could become heavily skewed at this initial step. We wanted to sample as many genres as possible in order to create an accurate high level view of musical changes over time. We also needed to ensure our data had a sufficient number of songs from all time periods for accurate analysis.

The Million Songs Dataset[6] is ideal because it captures many of the most popular songs from the mid 1920's to 2011 and is not heavily biased to specific genres. The FAQ section of the dataset's website details the parameters of how they choose which songs to include:

"

1. Getting the most 'familiar' artists according to The Echo Nest, then downloading as many songs as possible from each of them
2. Getting the 200 top terms from The Echo Nest, then using each term as a descriptor to find 100 artists, then downloading as many of their songs as possible
3. Getting the songs and artists from the CAL500 dataset
4. Getting 'extreme' songs from The Echo Nest search params, e.g. songs with highest energy, lowest energy, tempo, song hotness, ...
5. A random walk along the similar artists links starting from the 100 most familiar artists

The process used to obtain the million songs for this dataset uses familiar artists and the top 200 terms from Echo Nest, which is a music research company that focuses on collecting data and intelligence. This, along with the fact that they perform a random walk along similar artist links ensures that our data will cover a broad spectrum and contain many of the most popular songs over time. This makes us confident that our song set is not heavily skewed toward certain genres or time periods.

Once we obtained the Million Songs Dataset [6], we used these songs to query the Spotify API endpoint to obtain our data. The endpoint allows you to query for albums, tracks and artists. It then returns, through OAUTH authorized HTTP GET requests, a JSON dictionary of data ranging from audio features, album and track data, and market data. To query the Spotify API with our song name data from the Million Songs Dataset [6], we used the python requests module on the Spotipy API (Spotipy); Then the python SQLite3 module to store our data.

The data we collected for each song included:

- name
- URI (Spotify-unique ID, can be found on Spotify by right clicking a song, going to share)
- release date
- explicit or clean
- key
- tempo
- mode
- danceability
- valence
- energy
- loudness
- speechiness
- acousticness
- instrumentalness
- liveness
- duration

Data Preprocessing and Cleaning

After getting the song set, we had to clean the data so that our Spotify searches would be more successful. For example, underscores were used as commas in the song names and semicolons were used to separate multiple artists for a single song. We replaced these sorts of values with commas using some basic Python scripts. These

scripts also removed some extraneous information from the song names. Things like (LP Version) and artist features (eg. feat Rihanna) were removed to make the searches more generic.

Once the song list was cleaned and in such a format to derive results from the Spotify API, we could then write a Python script to first search for the name and retrieve the distinct id for each song. To do this, we used a package called 'Spotipy', which included functions to easily search based off of a string (not unlike how a regular user search for a song in the normal UI). From this search function, we could then pull the list of IDs from songs that returned results. While the data cleaning of the song list absolutely allowed the search function to return more results, there were a few examples in which the song existed in Spotify, yet the search could not reconcile between formatting. In these cases, a solution could be having a human manually research into the reasons for the search failing, and inputting the ID manually. However, with our dataset, these cases were few enough and our search returned more than enough data that we could ignore these cases and save many hours of manual labor.

One challenge we encountered while doing the initial search to get the IDs was rate limiting from the API. Spotify handles rate limiting a bit differently than some APIs; instead of limiting the number of calls you can make per day, they use a time delay if you make too many requests too quickly. Thankfully, the Spotipy package handled parsing the response headers and sleeping based on the amount of wait time they specified. Still, this dramatically increased the time it took to gather the data. The script had to make a request for each of the million songs because you can only search on a single string query at a time (vs. searching on 50 song IDs at a time). It ran for approximately 48 hours before finally finishing.

Once we managed to pull our list of distinct song IDs from Spotify, we could be sure that every song in this list actually existed, and could in turn pull the audio analysis data that Spotify provides. We could then use Spotipy again, using the built in function to pull audio features, with the attributes listed above.

For simplicity, all song attributes were inserted in a single table, which removed the need for any SQL join logic which would impact query performance. As well, some songs were not found on Spotify due to discrepancies in the songs' names including extra artist features, added hyphens and parentheses, searching for the remixed version when only the original is available, and others. When this occurred, we excluded the song from the dataset.

Once we had audio features information per song ID, we needed to again clean the data to ensure the data was appropriate for analysis. Immediately it was obvious to remove any duplicate IDs, which removed around 120k rows. These duplicates were detected by running our song data through a dictionary hashed by song ID, and from there, added to our SQLite database. Though we didn't initially realize, some other aspects of the data pulled from Spotify were suspect, which were uncovered as we began analysis. Our (incorrect) assumption, initially, was that Spotify standardized the data themselves, and that once we had data in the database everything would be fine for analysis. Unfortunately we realized that there were either some errors in the algorithms they ran, or those who submitted songs to Spotify submitted some data incorrectly, so our data had to be cleaned once again.

Luckily, we could write a SQL script in order to pull only the data we wanted for analysis. In doing this, we could improve performance, instead of having to import all the data into a DataFrame, then cleaning it there. By using a SQL query to process the data beforehand, there would be no unnecessary data to pull, thus improving performance. We found that there was quite a large range of song lengths, ranging from songs that are 0 milliseconds long (impossible) and 1000 milliseconds (one second), to songs greater than an hour and a half. To resolve this, we only included songs that were longer than one minute, because we had doubts about the accuracy of the Spotify algorithms run on such a short sample. Additionally, we found that there were quite a few impossible dates on songs, such as the artist who made the year the same as the album name for some artistic reason, so we limited our dates to 1899 onwards. We also found that both time signature and tempo had impossible values, so they were removed in the query. As a catch-all for any other possible issues in our data, we also added clauses to remove nulls or any other possible missing value.

In order to facilitate certain analyses, we also converted certain attributes in order to ensure they were in the correct formatting, as well as analysing each attribute and storing

the categories in new columns. The reason for this makes it easier to predict classifications, as opposed to attempting it with continuous data (which can be pretty fine grained, with decimals). Additionally, these categories provide a context for how each attribute should be interpreted, as it becomes difficult to judge relatively what each number means.

After all the duplicates were removed with the initial data cleaning, we ended up with about 690,000 songs. With the SQL query, we had about 675,000 songs remaining, with about 81,000 distinct artists. Of these artists, notable names include: The Beatles, Bruce Springsteen, Bob Dylan, Elvis Presley, and Stevie Wonder- each writing about 130 songs in our dataset.

Data Analysis

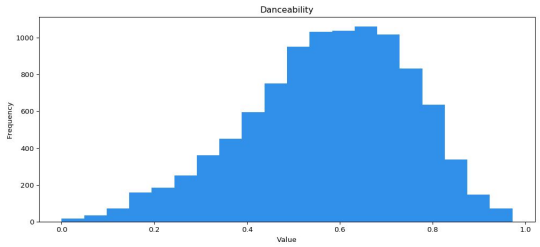
Our analysis was, as stated earlier, to identify past quantitative trends in musical pieces to possibly predict future qualitative trends.. A few of the main questions we aimed to answer were:

- Can we observe the impact of crises (e.g. 9/11, 2008 recession, etc.) in the attributes of songs in that time period? For instance, is the valence (happiness/sadness) or tempo of songs impacted by such events in following years? Alternatively, can we see the rise of these attributes for positive events (e.g. the economic boom of the 80's)?
- How has overall musical lyricism changed over time?
- Are musical with specific tempos, time signatures and keys more popular now than in the past? Are attributes of popular music consistent over time? Was there a point where it became consistent?
- Are there statistical trends in the foundational attributes of music, such as tempo, time signature, or key? Are these attributes correlated to one another?
- Naturally tempo and dancibility are highly correlated, what about other attributes? What musical attributes have strong correlations with one another?
- Can we predict if a song will be danceable based on its other attributes?
- If an artist were to release a song tomorrow, can we predict if the song will be popular based on certain attributes? (stretch goal)
- ~~Can we classify a song's genre based on its attribute values? (stretch goal)~~
    - The data we obtained from Spotify does not include genre information. Plus, the genre of songs can be subjective and has diverged significantly in recent decades. There are many more genres and subgenres today than there were in past times and this would increase the difficulty of classification.

Design & Implementation

To analyze our data, we made a variety of different graphs, gathered useful statistics from our dataset and ran multiple clustering and classification algorithms to find trends. Our graphs show how certain features have changed over time and regressions on the change of these features.

To predict danceability for songs based on their features, we took the continuous valued attribute for danceability and split it up into two classes, low danceability and high danceability. The danceability distribution for the population (all songs on Spotify) explains how we split the classes:



It's clear that danceability follows a roughly normal distribution centered around 0.6. With this information, we decided to classify songs below 0.6 as low danceability and songs with 0.6 or above as high danceability. Using these classes, we were able to predict which class a certain song belonged to (since you can't predict the class of a continuous valued attribute) based on its features. We employed a logistic regression model as well as a k-nearest neighbors clustering model to generate predictions.

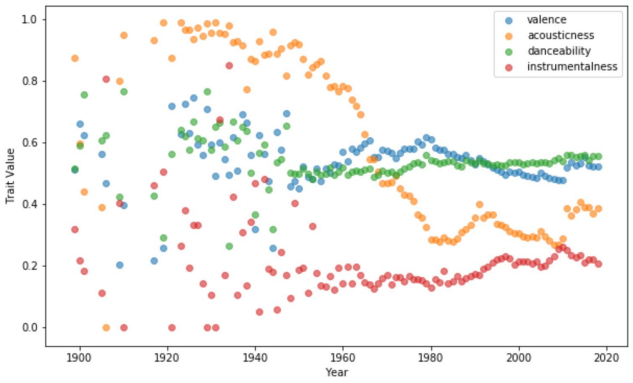
There are many data points that we had the ability to draw insight from because the Spotify API allows you to access pre-calculated song, album, and artist statistics from an almost infinite dataset of music. This could include, for example, tempo over time, danceability vs. tempo for different decades, or trends in speechiness (amount of times singer is speaking) and instrumentalness.

Evaluation

Temporal Trends

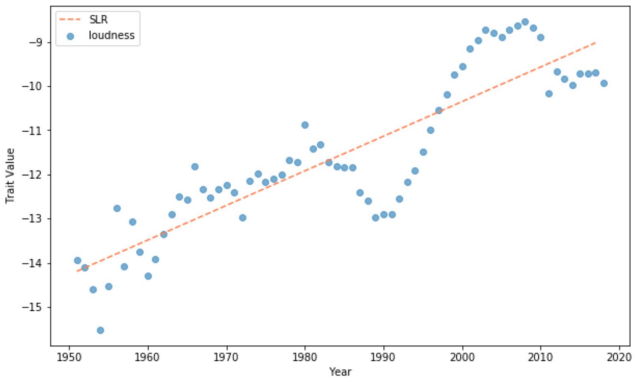
A bulk of our evaluation came from plotting traits in various ways and visually analyzing the graphs created. This yielded some interesting results, as we were often able to clearly see trends and patterns in music through time.

Our first method for plotting a trait temporally was to simply make a scatter plot with the trait value and release year for every song in the dataset. However, this proved ineffective as the data contained too many songs to reveal any sort of pattern or trend. Instead, we decided to average the trait value for each year, and plot that instead. For example, if we wanted to plot the danceability over time, we would take each year and average the danceability of every song in the set released in said year. This proved to be a much better method, as it revealed trends that are more true for the whole of the dataset. In the following graph, we plotted valence, acousticness, danceability, and instrumentalness on the same plot.

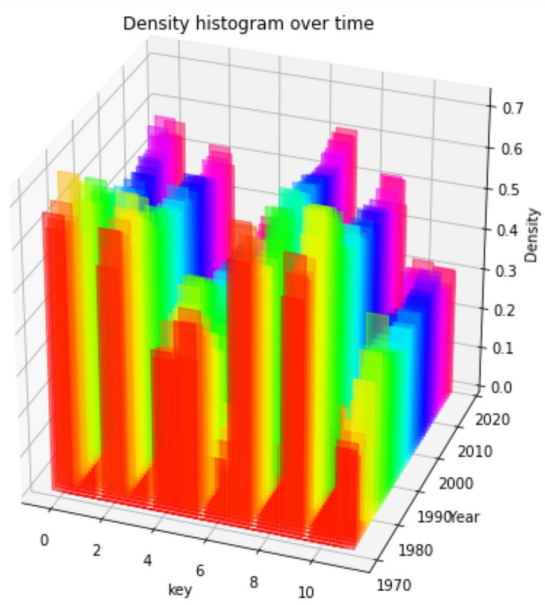


The data is rather sporadic before 1950, but this is because there are not many songs in our dataset from before that time, making the averages less predictable.

We also added some simple linear regression to this model which presents trends in a clearer way. In our graph of loudness, we confirmed some knowledge that we had before. That is, that loudness has steadily increased over time.



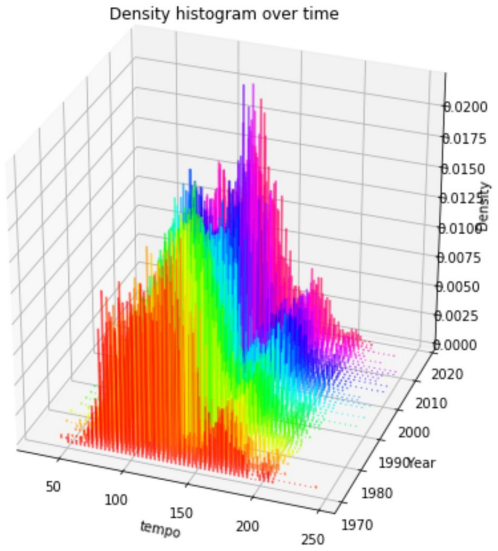
Another method of evaluating temporal trends was graphing the distribution of a trait over time. Instead of averaging each trait, we created a histogram of the trait for each year. This created a beautiful 3d graph that clearly shows some of the trends over time. For example, this is the image produced by graphing the key traits' distribution over time:



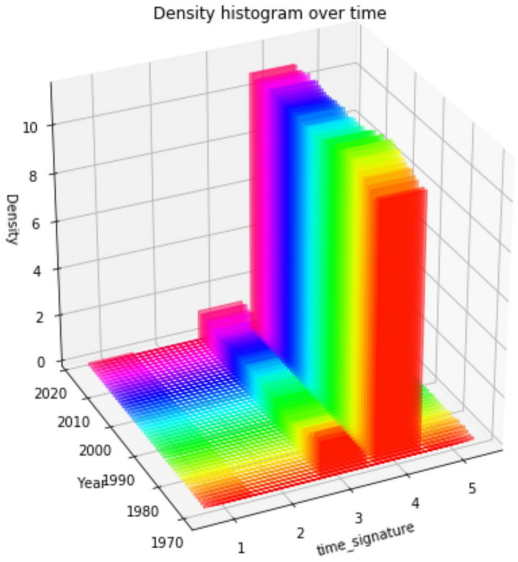
This graph is interesting, because it shows some clear spikes for some key values rather than others. These spikes may seem random, but they actually map to the notes of the C major scale. This suggests that songs in a key closer to C may be more popular than songs that are farther away.

We noticed a similar trend for tempo in songs. In the graph, there is a clear majority around the 100 bpm range, and another around 160 bpm. This shows that the majority of

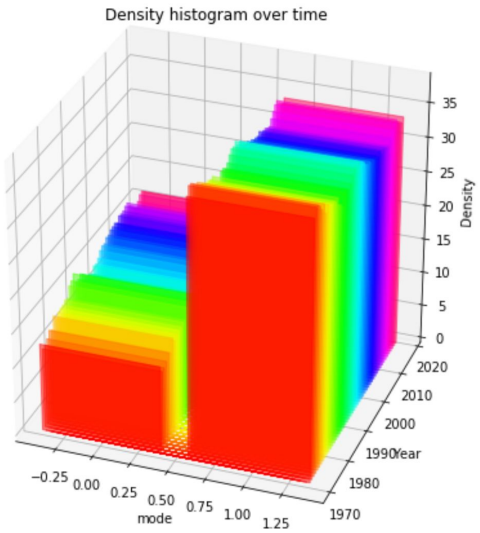
our songs fall near these tempos, meaning most popular songs.



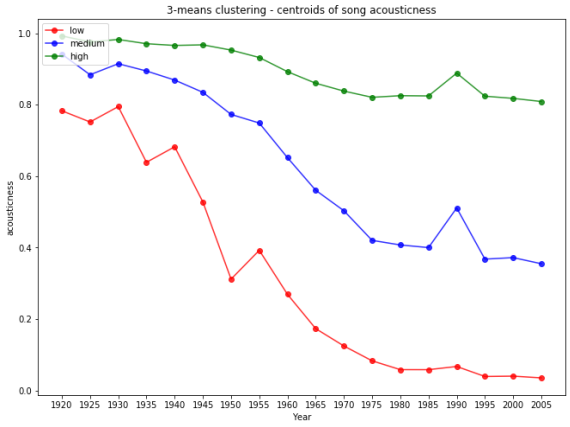
As might be expected, we found that the vast majority of our songs were in a 4 / 4 time signature, with the rest of them in a 3 / 4 time signature. In this graph however, you can see that the 3 / 4 time signature has become slightly more popular in recent years, and is taking up a more significant portion of songs.

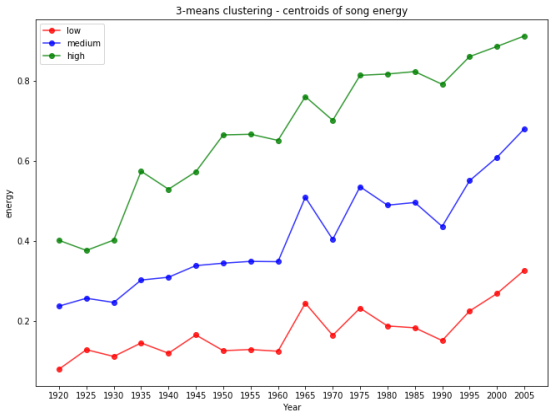


Similarly, songs in a major key are significantly more frequent than songs in a minor key. However, in recent years minor keys are becoming more popular than they were previously.



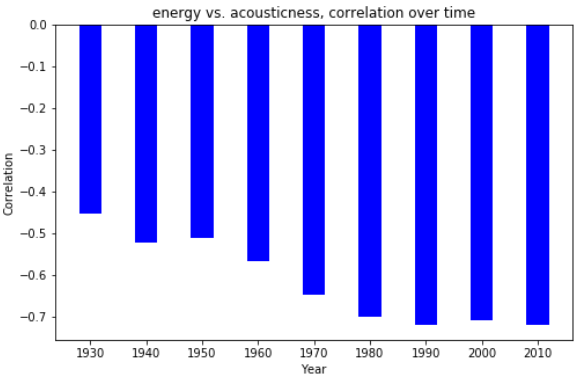
Clustering methods were also powerful in modeling a traits temporal change. We conducted a k-means clustering for all data points in a 5 year span for each trait. This provided us with a clustered average of each traits' low, medium, and high points over time. The graphs below, using the same method, show temporal changes in energy, acousticness, valence.





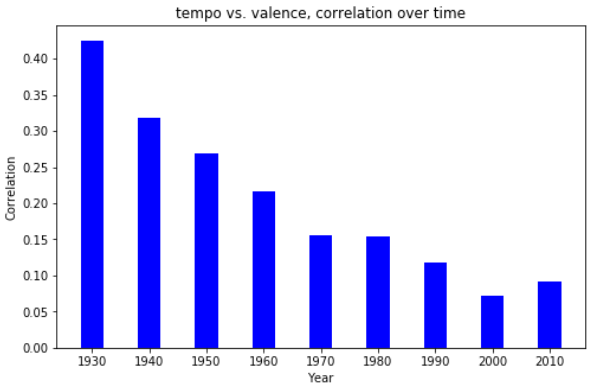
Clustering provided an advanced method of temporal analysis that gave us a more detailed evaluation of the data's average trend. The first graph of temporal clustering of acousticness shows an overall decrease in acousticness. More specifically, the low, medium, and high clusters lines, respectively, show higher rates of decrease. This shows that while a decreasing subset of songs *today* are heavily acoustic, there has been a higher decrease in the overall number of songs which show a high amount of acousticness. The second graph, similarly, shows an overall increase in song energy.

To accompany these findings, we wanted to if our one-trait analysis extended to 2 traits correlations. We plotted the correlation between traits over time for all trait combinations.



,

This graph shows shows an increasing correlation, or connection, between a song's energy (typically, energetic tracks feel loud, fast, and noisy) and acousticness. Combining these findings with findings from the previous graph of 3-means clustering on acousticness, its quickly noticed that while acousticness has trended down over the years, it has become increasingly connected to musical energy.

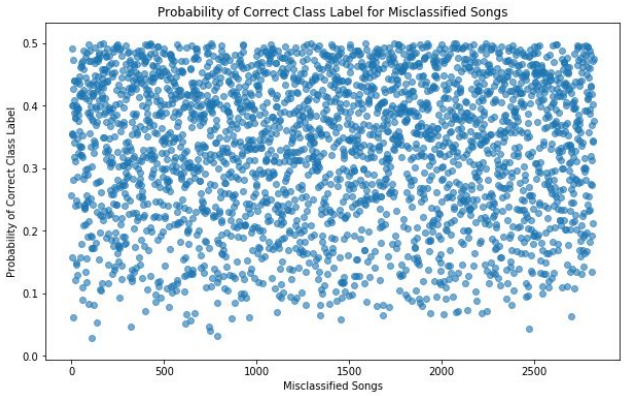


We can also see tempo and valence (a song's general level of happiness) have become less correlated to each other over the years. Our earlier findings showed that BPM (beats per minute) of around 100 are consistently the more frequent over the years. This could possibly show the tempo of song is being less frequently used as means for a song to be considered 'happy' songs. With an increase in song energy and decrease in acousticness, this could point to preconceived notions that typical 4-beat rock-n-roll songs, which are shown to be more static in their tempos, are being increasingly pushed out of the mainstream and replaced by electronically produced songs.

Classifying Danceability

As stated in the methodology section, we used both a logistic regression model and a k-nearest neighbors clustering model to predict the class label for danceability as either high or low. For the logistic regression, we began by using the Python package SKLearn to fit a binary logistic regression classifier using all of the features (besides the response of course). We found that when the song duration attribute is included, the predictions are much worse. This is most likely due to overfitting of an unnecessary attribute. The duration of a song doesn't say much about how danceable it is. After removing this feature, as well as other irrelevant features like artist, song name and ID, we arrived at a decent regression model. The attributes used in this model are release year, explicit, time signature, key, tempo, mode, energy, loudness, speechiness, acousticness, instrumentalness, liveness and valence. This model was made using all of the data points and when tested on random samples, correctly classified the danceability 73% of the time. To further determine the accuracy of our classifier, we made graphs to show the probability the

classifier gave to the correct class label for misclassified songs. When determining what class to assign to a song, probabilities are generated corresponding to each class. The class with the higher probability is then chosen. We made a function to see what the probability was assigned to the correct class for songs that were incorrectly classified.



As shown by the graph, the probability of the correct class follows a sort of gradient from 0.5 to 0. This makes sense because more misclassified songs should have close to a 0.5 probability associated with the correct class. The graph becomes more sparse below ~0.3, which supports this.

The next step was to determine what subset of those 13 features was most important for determining the danceability. To accomplish this, we ran a recursive feature elimination algorithm using our initial classifier. We found that about half the features in our initial classifier were not significant. The reduced model consists of time signature, energy, speechiness, acousticness, liveness and valence. After testing our reduced model on random samples from our dataset, we found that it was also around 72-73% accurate.

One interesting observation is that the tempo attribute was not included in our reduced logistic regression model, when we initially thought it would be the most important attribute in determining danceability.

After the regression, we used a k-nearest neighbors clustering model to see if it performed any better at classifying danceability. This was also accomplished using SKLearn. We used a training set that consisted of a random 80% of our song data and a testing set that used the other 20%. We used a minkowski distance metric to determine the neighbors since our data has a significant amount of features. We found that the using all the features from our full logistic regression classifier performed very badly (only about 30% accuracy) when used in a knn classifier. This is due to the curse of dimensionality. Since many of those attributes are irrelevant, it only serves to spread out the clusters and make the neighbors farther apart. When we used the reduced feature set with knn, it performed similarly to our logistic regression approach. Specifically our knn approach was 71% accurate when tested.

The most interesting result from the danceability predictions is the set of features which are most significant. In particular, we found it curious that speechiness and liveness were important in predicting danceability. One might think that both lyrical music and non-lyrical music would be similarly danceable, but this is not the case. Similarly, whether a song was performed live or not doesn't seem like it would have an impact on danceability either. Features like energy, time signature and acousticness make more sense in the context of danceability.

Discussion

Throughout the course of our project, we learned a number of things about working with large datasets and how to properly analyze them. We made a number of mistakes. For one, in our initial pull of the Spotify API we were limited to only one query at a time. We knew this would take a while to run, but we underestimated that time dramatically. The script took 48 hours to run, and we learned a lot about rate limiting in the process. Luckily, the Spotify API allows you to retry a query after only 1 second. This made things significantly faster than if they only allowed a certain amount per day, and we had to wait 24 hours between queries.

We also spent a significant amount of time attempting to implement the Apriori algorithm to find significant itemsets. We did this in an attempt to find if there are traits in songs that occur frequently together, but it did not prove useful in analysis. The algorithm was successful, but it did not show us any useful information about our dataset. This was a useful lesson, as it caused us to be more careful about what types of analysis methods we implemented, and to make sure the methods would be applicable to our data.

There were a lot of limitations in the work we could get done. For starters, there were a few traits in our data with consistently strange numbers. For example, the duration_ms attribute did not seem particularly accurate, and yielded some strange results. Additionally, tempo was

inaccurate on some occasions due to the computer unknowingly doubling or halving the true tempo of the song. This made it a little difficult to do some analysis, but in many cases we simply ignored the broken data.

Another problem was the release year in our dataset. Because of things like remastered albums, the release year might be drastically different than the year the song was originally released. For example, there were some Bach songs that, despite being initially written hundreds of years ago, had a release date of 1995 due to a new recording of the song being released. This is a bit of a problem with our data, but we did not find it to significantly impact our results.

One of the issues that impacted our project significantly was the lack of a popularity attribute. We initially thought that we would be able to easily get the popularity of a song from Spotify and based many of the questions we wanted to answer around that. For example, we wanted to predict if a certain song would be popular or not based on its attributes, or if popular songs tended to have certain tempos or time signatures on average. Unfortunately, we were not able to do this due to the lack of data on popularity. The major takeaway here is that it's critical to get to know your data and figure out what is feasible before formulating a plan of analysis.

In the future, we could attempt to retry some of our analysis on a better and more accurate dataset. To do this, we may have to create algorithms to calculate the traits ourselves in order to ensure they are as accurate as possible. In addition, we could select a more representative sample of popular music, rather than simply using the songs that are in the million song dataset. While the million song dataset has a huge amount of music in it, it contains some songs that are anomalous and possibly do not belong in a study of this type.

Much of our research can be directly applied to people creating music in the music industry today. It is extremely useful to know what choices to make in the song creation process to ensure the song is as popular as possible and therefore can turn a big profit.

Conclusion

Over the course of finding and collecting our dataset, we have learned a number of lessons. One of the biggest lessons is how inconvenient data rate limiting can be. In our initial search through the Spotify API, we were only allowed one query at a time, which ended up causing our first script to run for 48 hours. In our second query of the Spotify API, we were able to take advantage of a different API request method that allowed us to query 50 song ID's at a time, which greatly reduced the run time.

We also learned that data cleaning is a much more important step in the process of retrieving a dataset than we first anticipated. Throughout multiple steps in the process, there were many extra bits of unnecessary or redundant information that needed to be filtered out. Had we not caught these redundancies, it is likely our final analysis would be dramatically skewed. It was particularly important to make sure this step was done correctly in order to make sure results could actually be pulled from the Spotify search.

Our key methods for analyzing the data were making graphs, both in two and three dimensions. These graphs helped to visualize how different trends changed over time, which was one of the main goals of our project. We used k-means clustering for 5-year time spans to find low, medium and high cluster points for various attributes in our data over time. This allowed us to get a more detailed evaluation of how features changed over time than simply using the mean of a feature over the years. Logistic regression and K-nearest neighbors classification was also an important method we used to draw knowledge from our dataset.

Some key findings from the graphing stage is the discovery of most common keys and tempos. It seems likely that all keys and tempos would have had a roughly equal representation, but instead we found that there were some groups that were much more likely than others. Interestingly, they distribution of keys followed notes from the C-major scale. Tempos around 100 bpm were also by far the most common.

We also discovered that changes in the energy and acousticness of songs, combined with decreased correlations between tempo and valence, support preconceived notions that in recent years, 4-beat rock-n-roll style songs, which show heavy correlations between tempo and valence, are being increasingly pushed out of the mainstream and replaced by electronically produced songs of varying tempos.

Our project also found the important attributes for prediction a song's danceability. We found that six features, namely time signature, energy, speechiness, acousticness, liveness and valence can predict the danceability with an accuracy of over 70%.

References

[1] The Echo Nest. 2013. How Music Has Evolved in the Past 70 Years. (December 2013). Retrieved October 11, 2018 from https://gizmodo.com/how-music-has-evolved-in-the-past-70-years-1485770090

[2] Shane Snow. 2017. This Analysis of the Last 50 Years of Pop Music Reveals Just How Much America Has Changed. (February 2017). Retrieved October 11, 2018 from https://contently.com/2015/05/07/this-analysis-of-the-last-50-years-of-pop-music-reveals-just-how-much-america-has-changed/

[3] Madison Smith, Kelly O'Donnell, Nimish Garg, Andrew Procter, and Kasey Jones. 2017. Songs Over the Decades – A Text Mining Analysis. (January 2017). Retrieved October 11, 2018 from https://datacolumn.wordpress.ncsu.edu/blog/2017/01/27/songs-over-the-decades/

[4] Nie Yi-Bo. 2012. Data Mining applied to Music Style Classification. (2012), 1–6.http://ijssst.info/Vol-17/No-2/paper19.pdf

[5] Anon. 2018. r/dataisbeautiful - Songs have gotten louder over time [OC]. (April 2018). Retrieved October 7, 2018 from https://www.reddit.com/r/dataisbeautiful/comments/88q0d1/songs_have_gotten_louder_over_time_oc/

[5] Anon. Get Audio Features for a Track. Retrieved October 11, 2018 from https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

[6] LabROSA, The Echo Nest, Columbia University New York, 2011, "Million Song Dataset" https://labrosa.ee.columbia.edu/millionsong/

Appendix

Honor Code Pledge:

Honor Code: "On my honor, as a University of Colorado Boulder student, I have neither given nor received unauthorized assistance."

Individual Contributions:

Matthew Menten: API work, database creation, initial data cleaning, logistic regression, knn classification

Kieren Ng: SQL queries, API work, data cleaning, apriori algorithm implementation

Braden Holmes: Temporal plotting and 3d histograms

Tanner O'Rourke: API work, K-mean clustering, correlations over time