Spotifying Trends in Popular Music

Date

2017/12/04

Group Members

- Anastasia Vela
- Joshua Asuncion
- Kaiwen(Kevin) Pang
- Steve Hwang

Link to GitHub: https://github.com/anatasiavela/Spot-the-Future

Introduction

Have you ever wondered what is common about top tracks every year? Are you interested in using data to predict the next popular songs and the songs that are most likely to win Grammy Awards? In this project, we have used Spotify API and collected audio features such as danceability, energy, accousticness, and more for Billboard top 100 songs from 2012 to 2016. We cleaned and created visualizations that show very interesting trends among top 100 tracks in each year and time-series connections across years. In addition, we made predictions using features from previous years to predict the “next popular song”.

Related Work

In the past, people have utilized Spotify’s API for interesting projects such as automatically creating playlists by comparing and categorizing similar songs. Another spotify user was able to pluck random songs from his coworkers’ playlists and compile them into a master playlist for the entire office. This master playlist provided a way for the office workers to connect to one another through their common or uncommon

interests in music. In addition, the Spotify corporation has used their workflow manager software, Luigi, to accurately predict 4 of the 6 Grammy winners by taking into account Spotify users' listening habits. Our work is different because we are using song features rather than user's listening habits to predict the next big hit song.

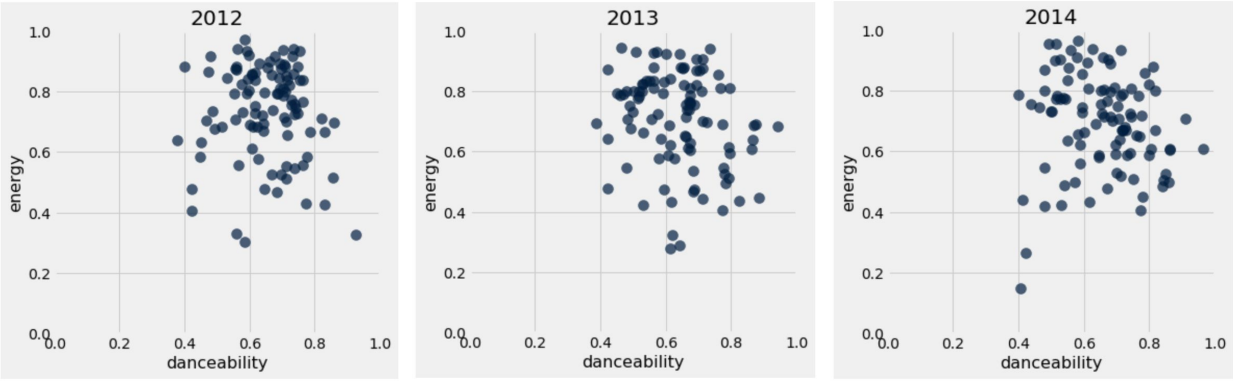
Data

The dataset is top 100 songs each year on Billboard for the past five years. For each song, we collected 13 audio features. Some features are numerical variables ranging from 0 to 1 such as energy and danceability while some features are categorical values 0 or 1 such as mode.

In our machine learning predictor example, we used the 500 songs as a training set. In addition, we collected top 50 songs with audio features in 2017 and used it as a test set to predict one song from 2017 with closest average features calculated using top 100 songs from 2012 to 2016.

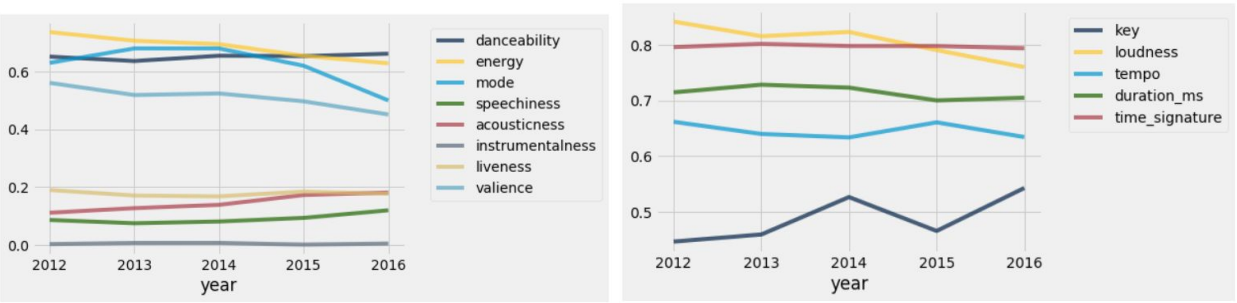
Visualizations

Interactive Time Scaled Scatter Plot using Python



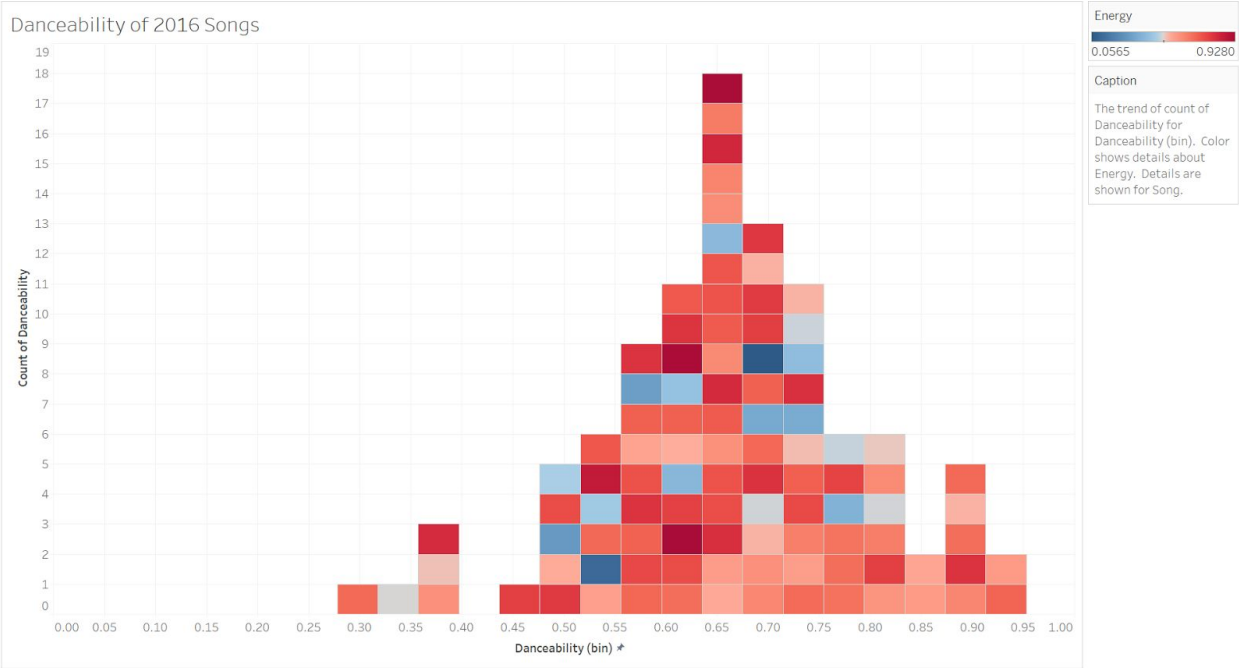
2 columns of the song features data table (danceability and energy) are chosen to be the x- and y- axes of the graph. Then each of the 100 top songs is plotted depending on their respective feature. A dial can be switched to change the presented year. However, screenshots were captured instead to handle the lack of interactiveness. There is a slightly present trend that energy and danceability decreases as time progresses. Similar trends can be analyzed by changing the 2 features on the x- and y-axes.

Line Graph

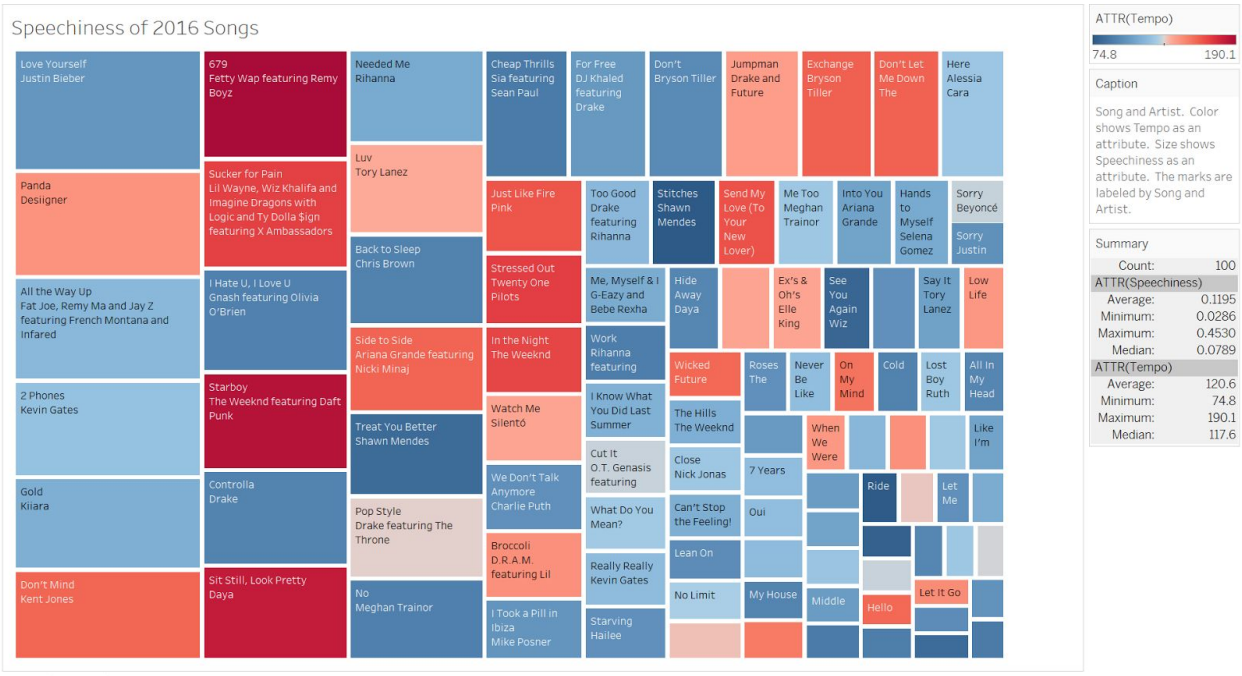


The data present in the graph to the left is scaled on a range of 0 to 1. 0 being the least of the feature and 1 being the strongest of the feature. From each year, an average of the feature was calculated from the top 100 songs and plotted. However, with the features on the right, the data is scaled in an arbitrary way, so the data was standardized depending on each feature data from every year then put into a scale of 0 to 1. With this, trends can be seen as time goes on.

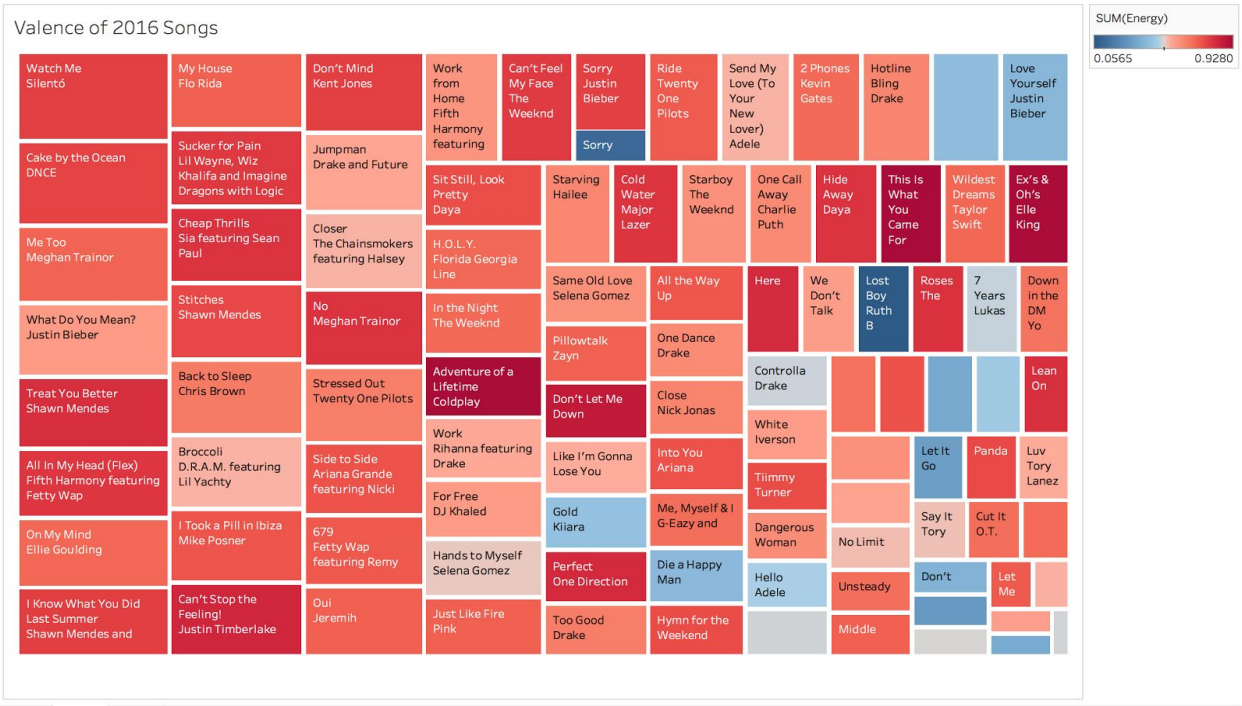
Tableau Visualizations



The distribution of danceability follows a distribution that is similar to a normal distribution while energy level spreads out and does not show a clear pattern of distribution.



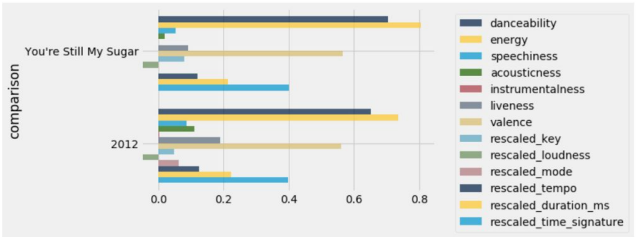
All top songs have low speechiness levels with a maximum of 0.453. This means that most top songs have more music content than speech content. Songs with values below 0.33 most likely represent music and other non-speech-like tracks. Interestingly, we observe songs with higher speech levels have high tempos as well. This is represented in red. One possible conclusion of this finding is that some songs especially rap songs with high speech content have faster tempos as well.

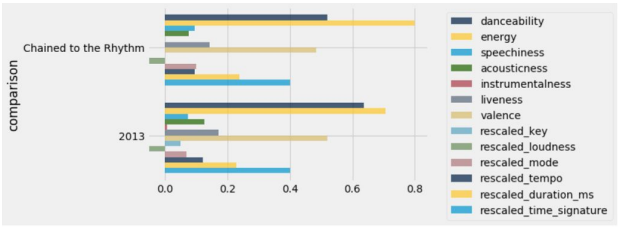


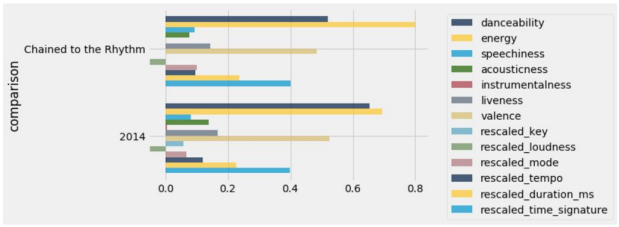
From this heat map, we observe that high valence (happy and cheerful) songs tend to have high energy. The correlation between valence and energy is positive.

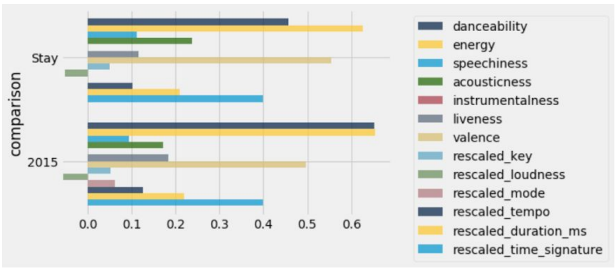
For the interactive Tableau visualizations, visit Tableau Public:
https://public.tableau.com/views/Spotify_10/Sheet1?:embed=y&:display_count=yes

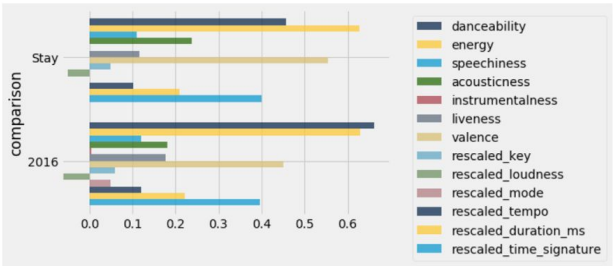
Predictor





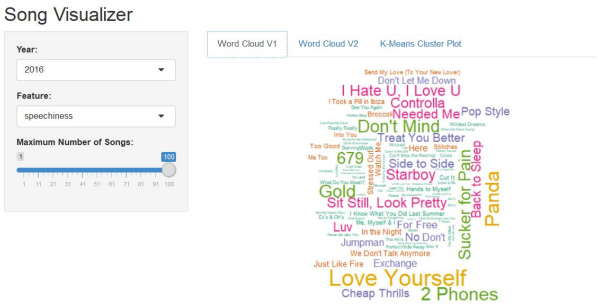


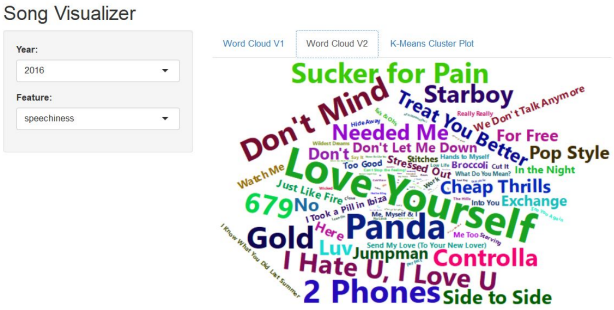




In the bar plot, the features of the predicted most popular song of 2017 are the top bars and the average features of the year used for comparison are the bottom bars, where each bar is the value of a feature. The data was rescaled such that all the features fit on a 0 to 1 scale. The bar plot shows how different the predicted song's features are from the features of the comparison dataset. Using the 2012 dataset, "You're Still My Sugar" was predicted as the most popular song. With 2013 and 2014, "Chained to the Rhythm" was the predicted song. With 2015 and 2016, "Stay" was the predicted song.

Shiny App





The Shiny app features two different word clouds and a k-means cluster plot. The first word cloud in the app takes in a selected year and feature and creates a word cloud where the largest songs have the largest values for that feature. In addition, the user has the ability to change the maximum number of songs plotted in the word cloud. The second word cloud in the app is similar to the first, except with different artwork and without the ability to change the maximum number of songs plotted. The k-means cluster plot allows the user to choose a year and two features, plot the songs as a scatter plot, and group the songs together by clusters. The app gives the ability to change the number of clusters from 1 to 9, in addition to the ability to hover over a point and retrieve the point's song, artist, and feature values.

Link to Shiny App:
https://joshasuncion.shinyapps.io/song_visualizer/

Conclusion

As displayed from the visualizations, we found the energy of the top 100 Billboard songs goes down from 0.737 in 2012 to 0.628 in 2016, while danceability slightly increases from 0.651 to 0.662. In addition, we found that the loudness for the song decreases from 0.842 to 0.760. This implies that the trend for the more popular songs is is changing to less energetic, but more danceable songs, which is peculiar because it would be assumed that the more energetic a song is, the more danceable it is. Almost all top songs have a 4/4 time signature.

We noticed that more top songs are in the minor key (about 50% now) implying that even though minor keys are sadder, they can represent positive songs such as “Happy” by Pharrell Williams. Pharrell’s song is in the minor key, but has still gained much popularity.

Final Thoughts

One challenge that we faced: while we were collecting data, some tracks share same names with other songs by different artists and/or different versions. The code returned an error message because the API did not know which value to return. In order to resolve this, we found that Spotify assigns a unique ID to each track. Therefore, we decided to manually collect ID for each song by inserting an additional column and used ID to extract audio features.

Another challenge was we experienced some difficulties in deciding which visualizations to make and what tools to use because we didn’t have a clear question in mind. The whole project is based on curiosity and we learned by messing around with the data.

We had a lot of fun working on the project from collecting data to exploring different options and tools to make visualizations. Although we hit roadblocks when we first tried to use Facebook API to extract mutual friends data for our first project idea, we enjoyed the process of exploring other interesting options and coming up with this new idea. We used a variety of tools including Python, R, and Tableau to visualize our data

and have found interesting patterns and trends that can be used to explain real world phenomenons about popular songs.

References

We used the “spotipy” library to access the Spotify API in python format. In addition, we used “spotipy” to extract data from Spotify. We consulted the Data 8 library to create metaplots. Lastly, we used R to create our word cloud, which we used to visually compare the differences in songs between years.

1. https://github.com/plamere/spotipy
2. https://developer.spotify.com/web-api/get-several-audio-features/
3. https://www.billboard.com/charts/year-end
4. https://www.inferentialthinking.com/
5. http://spotipy.readthedocs.io/en/latest/#module-spotipy.oauth2
6. https://shiny.rstudio.com/gallery/word-cloud.html
7. https://shiny.rstudio.com/gallery/kmeans-example.html
8. https://shiny.rstudio.com/gallery/plot-interaction-basic.html