What's the Vibe

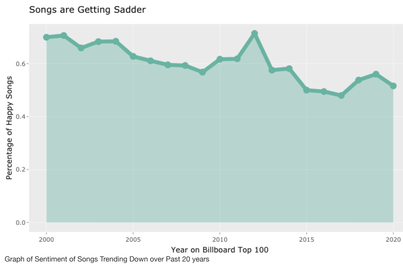
Deen Amanat, Mohsin Butt, Tianna Couch

12/14/2021

The purpose of our project was to create a model that could analyze whether a song should be classified as "happy" or "sad", and then use that model to analyze how the general sentiment of popular music in the United States has trended over the past 20 years. With depressions, pandemics, record economic expansions and tremendous social change, the United States has experienced some of the most unique events in its economic and social history over the past 20 years. Throughout this period, we want to investigate whether songs have become happier, sadder, or trended up and down along with the events playing out on the national stage.

In order to do so, we began by collecting a dataset of happy and a dataset of sad songs across different genres in order to train our random forest classification model. Diversity of genre was important in order to create a model that was robust to changes in the popularity of genres over different time periods. For example, hip hop music is the dominant genre nowadays, but in the early 2000s the charts were dominated by pop punk songs. By confirming the diversity of genre within out dataset, we ensure that our model will be accurate for our entire period of interest. We created these datasets by pulling user-created playlists from the Spotify API that contained either happy or sad songs. Working with the API helped to develop our knowledge of web scraping and made us aware of the vast amounts of interesting data that are out in the wild, not sitting neatly stored in databases. Training the model off these songs, we achieved a test accuracy rate of 75%.

After building the model, we wanted to use it to analyze how sentiment of popular songs has trended over the past 20 years. We used the billboard hot 100 charts of each year since 2000 as a proxy for popular songs in that year. We compiled that data by pulling billboard hot 100 playlists of each year from Spotify and fed them into the model, yielding the following result:



Graph of Sentiment of Songs Trending Down over Past 20 years

As you can see, there is a clear trend of songs getting sadder over the past 20 years. This might be related to the rising depression rates that we have witnessed over the same time period as social media has become more prevalent. Interestingly, we see the strongest rebound in sentiment during the recovery from the great recession. This might reflect the general positive notion at the time after coming from such a difficult period of economic hardship.

This study could have been significantly improved had we had access to lyrics data in order to run a sentiment analysis of our traning dataset that would factor into our model. However, we ran into copyright issues when attempting to scrape data from genius (a popular lyrics database) and as such were forced to abandon the idea. In addition, the study could be further improved by creating more classification groups besides happy and sad; after all, it is obvious that not all songs fit neatly into one category or the other!

An ethical dilemma that we confronted while conducting these analyses builds off of the latter suggestion for improvement in that art is not meant to be classified into boxes and neatly labelled, which is inherently what a machine learning algorithm does. There is so much nuance, passion, and energy that is lost when we assign a label of merely "happy" or "sad" to a song. While this analysis is certainly fun, it is by no means a definitive summary of all the diverse feelings and emotions that listeners experienced when hearing these songs at the peak of their popularity.

```
track_list_happy <- map_dfr(happy_playlist$value, get_playlist_tracks)
track_list_sad <- map_dfr(sad_playlists$value, get_playlist_tracks)
```

Once we gathered the data, we want to wrangle and clean it so that we have our variables of interest in a manageable format. In the end, we gathered 949 happy songs and 1235 sad songs.

Our variables of interest are: Tempo: The BPM of the song. Faster, more higher tempo songs are likely happier. Valence: Spotify's proprietary metric of song positivity on a scale of 0 to 1. Danceability: Spotify's proprietary metric of song danceability on a scale of 0 to 1. Mode: Whether a song is in the major key or the minor key. Songs in major key are more likely to be happy than those in minor key.

We wanted to scrape lyrics data as well in order to calculate sentiment scores for each song, however we ran into copyright issues with scraping the data. Therefore, we decided not to pursue this route.

```
SONG_data <- cleansing_song_data(track_list_happy, track_list_sad)
```

With our data, we wanted to look at some descriptive statistics for our variables of interest.

For happy songs: Mean tempo: 121.8637 Mean valence: 0.5846365 Mean danceability: 0.6518430 Mean mode: 0.7407798

For sad songs: Mean tempo: 117.5513 Mean valence: 0.3169351 Mean danceability: 0.5655433 Mean mode: 0.7157895

We were also interested in whether the variables are statistically different between each of the groups, and we found that each variable is in fact statistically different except for mode. Because of this, we initially thought to exclude mode from our model, however including it improved our performance in the end so we decided to retain it.

happy	mean_tempo	mean_valence	mean_dancebility	mean_mode
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
N	117.6243	0.3181097	0.5662387	0.7162379
Y	121.8637	0.5846365	0.6518430	0.7407798

2 rows

```
##
##  Welch Two Sample t-test
##
## data:  tempo by happy
## t = -3.4723, df = 2137.8, p-value = 0.0005262
## alternative hypothesis: true difference in means between group N and group Y is not equal to 0
## 95 percent confidence interval:
##  -6.633545 -1.845077
## sample estimates:
## mean in group N mean in group Y
##        117.6243        121.8637
```

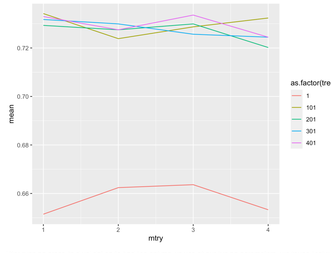
```
##
##  Welch Two Sample t-test
##
## data:  mode by happy
## t = -1.2827, df = 2068.3, p-value = 0.1997
## alternative hypothesis: true difference in means between group N and group Y is not equal to 0
## 95 percent confidence interval:
##  -0.06206352  0.01297987
## sample estimates:
## mean in group N mean in group Y
##       0.7162379       0.7407798
```

```
##
##  Welch Two Sample t-test
##
## data:  valence by happy
## t = -29.72, df = 1854.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group N and group Y is not equal to 0
## 95 percent confidence interval:
##  -0.2841159 -0.2493384
## sample estimates:
## mean in group N mean in group Y
##       0.3181097       0.5846365
```

```
##
##  Welch Two Sample t-test
##
## data:  danceability by happy
## t = -13.987, df = 2131.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group N and group Y is not equal to 0
## 95 percent confidence interval:
##  -0.09760651 -0.07360199
## sample estimates:
## mean in group N mean in group Y
##       0.5662387       0.6518430
```

After collecting and analyzing our data, we build a random forest classification model with mtry and # of trees as our tuning parameters.

```
SONG_rf_tune %>%
  collect_metrics() %>%
  filter(.metric == "accuracy") %>%
  ggplot() +
  geom_line(aes(color = as.factor(trees), y = mean, x = mtry)) # trees as legend var
```



After running cross-fold validation to find the optimal values of our tuning parameters, we found that we achieved the best error rates when our mtry was 1 and our # of trees was 201.

```
plottable %>%
  mutate(correct = ifelse(.pred_class == happy, 1, 0)) %>%
  dplyr::summarize(accuracy = mean(correct))
```

accuracy
<dbl>
0.7540984

1 row

We achieved a test accuracy rate of 75%, meaning that we predicted the correct classification of a song 75% of the time.

Now that we have built the model, we want to use it to analyze how sentiment of popular songs has trended over the past 20 years. We use the billboard hot 100 charts of each year since 2000 as a proxy for popular songs in that year. Here, we compile that data by pulling billboard hot 100 playlists of each year from Spotify.

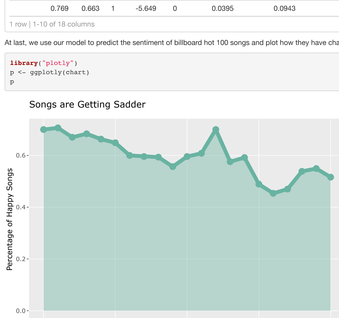
Just like we did for our training set, we then clean the dataset to make it workable and isolate our variables of interest.

danceability	energy	...	loudness	m...	speechiness	acousticness	instrumentalness	liveness	valence
<dbl>	<dbl>	<int>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0.769	0.663	1	-5.649	0	0.0395	0.0943	1.12e-06	0.133	0.46

1 row | 1-10 of 18 columns

At last, we use our model to predict the sentiment of billboard hot 100 songs and plot how they have changed over time.

```
library("plotly")
p <- ggplotly(chart)
p
```



Citations:

Charlie Thompson, Daniel Antal, Josiah Parry, Donal Phipps and Tom Wolff (2021). spotifyr: R Wrapper for the 'Spotify' Web API. R package version 2.2.3. https://CRAN.R-project.org/package=spotifyr

"Decade in Review: 2010s Was the Decade of the Bull." https://money.usnews.com/investing/stock-market-news/articles/2019-11-29/decade-in-review-2010s-was-the-decade-of-the-bull

Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL https://www.jstatsoft.org/v40/i03/.

Hidaka, Brandon H. "Depression as a Disease of Modernity: Explanations for Increasing Prevalence." Journal of Affective Disorders, vol. 140, no. 3, 2012, pp. 205–214., https://doi.org/10.1016/j.jad.2011.12.036.

Kuhn et al., (2020). Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles. https://www.tidymodels.org

Paul, Crystal (2019). "A Look Back at 10 of the Biggest Social Movements of the 2010s, and How They Shaped Seattle." The Seattle Times, The Seattle Times Company, https://www.seattletimes.com/life/a-look-back-at-10-of-the-biggest-social-movements-of-the-2010s-and-how-they-shaped-seattle/

Rich, Robert. "The Great Recession." Federal Reserve History, https://www.federalreservehistory.org/essays/great-recession-of-200709

Taylor, Derrick Bryson (2020). "A Timeline of the Coronavirus Pandemic." The New York Times, The New York Times, https://www.nytimes.com/article/coronavirus-timeline.html

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

Yihui Xie (2021). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.33.