Written Report

STA 210 - Project

Soy Nuggets - Felix Hu, Madeleine Jones, Isabel Siebrecht, Jason Yang

Introduction and Research Hypothesis

Spotify is a popular music streaming service that has 406 million monthly listeners as of 2021. **Billboard 100** is the de facto standard record chart for the US. The entity publishes weekly charts that rank songs by popularity.

The Spotify song popularity metric has major commercial implications. Record labels aim to accumulate as many Spotify popularity points in order to influence Spotify's recommendation system. That way, they can discover and accumulate more new listeners. (Eichler 2020)

With this in mind, our group aims to explore how certain characteristics of a song (both audio features and Billboard data) such as danceability, tempo, and the number of weeks a song was on the Billboard charts influence its Spotify popularity score, as determined by Spotify. More specifically, we 1) attempt to determine the most significant predictors and 2) try to use these predictors to predict Spotify popularity of a song. Record labels would find this analysis beneficial for their business because such insights could help them increase the streaming of songs under their label. We hypothesize that an increase in valence, energy, and peak position (by increase in position, we mean decrease in numerical value since position 1 is the best) on Billboard charts of a song are significant predictors that a song is popular by Spotify metrics.

- Eichler, Oskar. "Spotify Popularity- a Unique Insight into the Spotify Algorithm and How to Influence It." *Medium*, The Songstats Lab, 7 Oct. 2020, https://lab.songstats.com/spotify-popularity-a-unique-insight-into-the-spotify-algorithm-and-how-to-influence-it-93bb63863ff0.

Data description

Our data contains two csv files that will be the primary concern of this proposal: the billboard and audio_features files. This Billboard dataset came from the Billboard Weekly Hot 100 singles chart between years 1958 to 2020. The Spotify data was collected from the Spotify API in the year 2021.

The data was sourced from Data.World by Sean Miller, who scraped Billboard and the Spotify API for the relevant data.

We combined the two datasets by song id, which is the song title followed by the artist name. For each observation, `peak_position` is the highest position that the song's ever had on the Billboard charts, and `weeks_on_chart` is the total number of weeks that the song was on the charts. We list our full set of predictors and outcome variable:

- `peak_position`: historically highest position occupied on the Billboard Hot 100 chart
- `weeks_on_chart`: duration in weeks spent on Billboard Hot 100 chart
- `loudness`: how loud the track is in decibels
- `spotify_track_duration_ms`: length of song in milliseconds
- `energy`: value from 0.0 to 1.0 indicating track intensity (the closer to 1.0, the more energetic the track)
- `key`: key of track in standard pitch class notation
- `mode`: major or minor track (1 indicates major, 0 indicates minor)
- `liveness`: value from 0.0 to 1.0 indicating how live the track sounds (the closer to 1.0, the more live)
- `valence`: value from 0.0 to 1.0 indicating track positiveness (the closer to 1.0, the more cheerful the track)
- `tempo`: estimated tempo of track in beats per minute

For our outcome variable, we create a new boolean variable `is_popular`. To do so, we create a threshold between the popular and not popular songs; we choose to refer to songs with `spotify_track_popularity` score in the top quartile as "popular" while the rest of the songs would not be considered popular. The popular songs have a much higher chance of being added to recommendations and music radios than unpopular ones, giving financial incentive to our research. As justification for this choice, a song doesn't have to have a particular popularity score to be highly recommended, hence we are okay with predicting whether or not a song falls into a popularity score interval. Note: by our creation `is_popular` will be skewed.

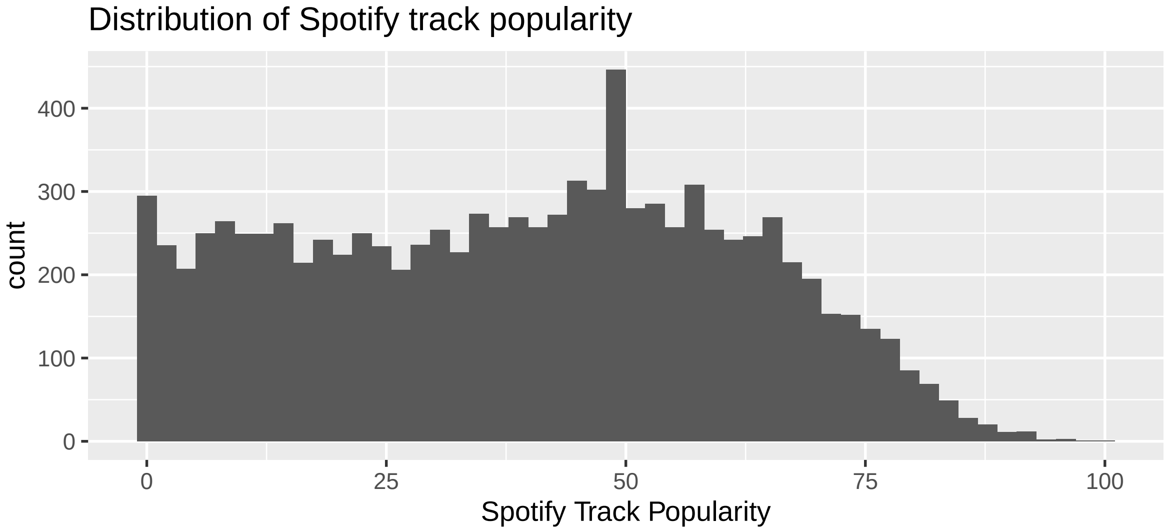
Part of our objective is prediction, so we first do a 75%-25% train-test split before looking at the data and creating the `is_popular` variable (we use the same threshold generated from the training set to create the variable in both the training and testing sets). The code is invisible in the PDF here but viewable in the QMD file. This protects us from data snooping.

Finally, our outcome variable,

- `is_popular`: a boolean that tells whether or not a song is popular (by popular, we mean in the top quartile of `spotify_track_popularity` scores)

Exploratory Data Analysis

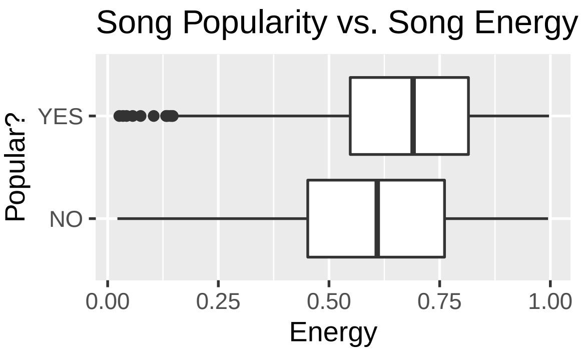
First we will look at the distribution of the original `spotify_track_popularity` variable.

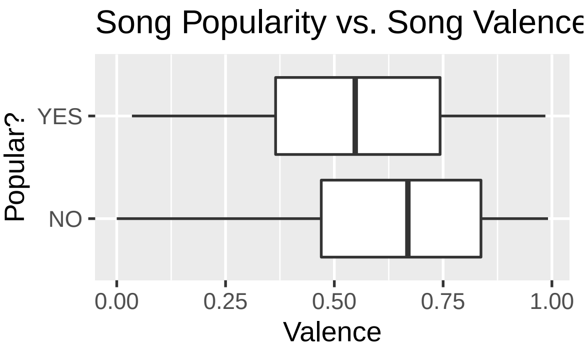


mean	median	IQR	std. dev
38.994	40	37	22.655

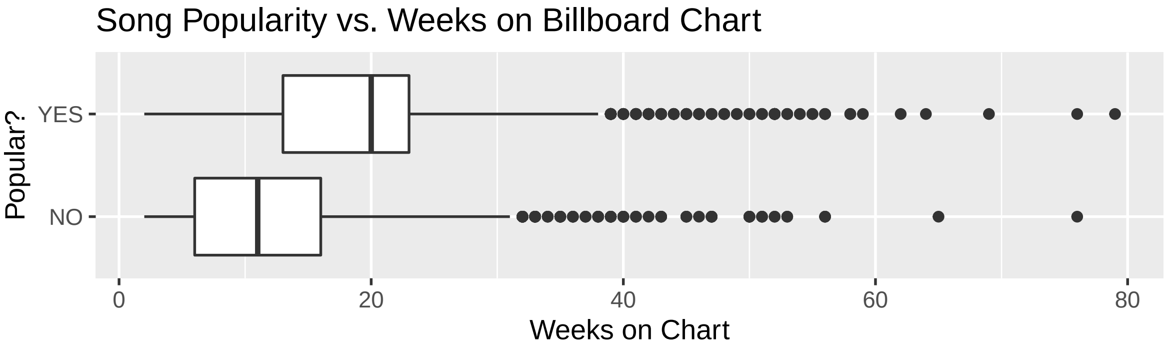
Looking at the graph and summary statistics, very few songs have high popularity score. This is a heavily right skewed distribution, making all the more important for us to be able to differentiate a song with high vs. low Spotify popularity. More importantly, it means songs that have been on the Billboard charts (and hence in our dataset) aren't guaranteed a high popularity score, a primary reason behind this paper.

To match our objectives, for the rest of the EDA, we will analyze the new popularity score variable `is_popular`.

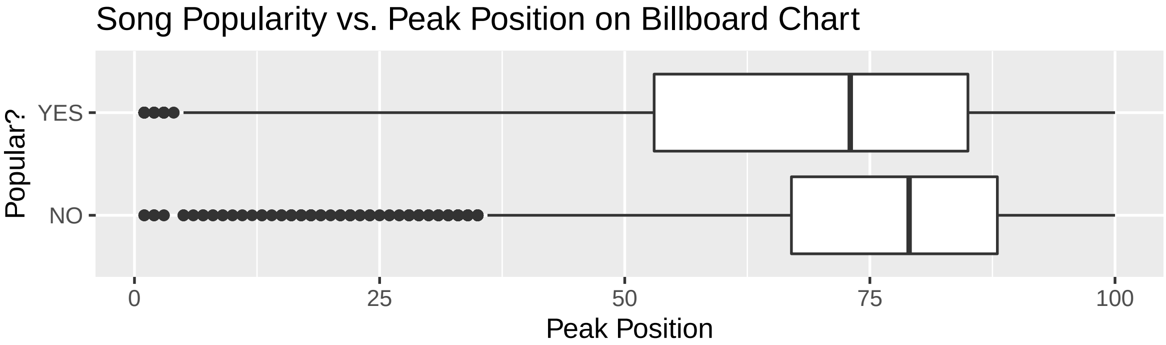




From the popularity score vs. energy plot, we can see that popular songs tend to see higher energy levels. We may infer that as song energy goes up, so does the likelihood of it being popular. The popularity vs. valence graph was an interesting find because it shows the opposite. We can see that popular songs tend to see lower valence (how positive the song is). In other words, at first glance, the happier the song, the less likely it is to be popular according to Spotify's scoring system - quite counterintuitive.



As we expect, our plot shows that songs that have spent more time on Billboard charts seem to be more likely to be popular by Spotify's metrics. There is a high number of outliers, but we don't expect the two to be perfectly correlated anyways. One possible reason that comes to mind is Spotify's popularity scores might favor really trendy songs that are viral for short periods of time whereas Billboard Chart songs tend to be more consistently trendy.



For peak position, we see that Spotify's popular songs generally have lower peak position in the charts, which is what we expect because charts are ordered inversely (i.e. #1 is the best chart position).

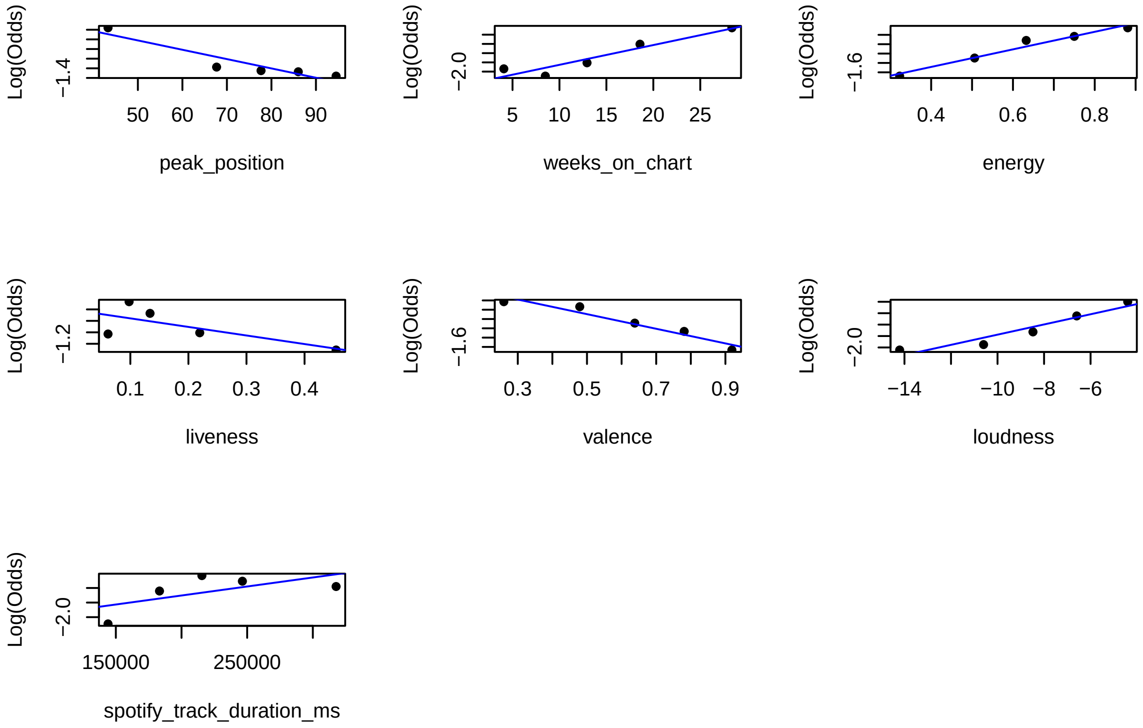
As a group, we were concerned about including the `weeks_on_chart` and `peak_position` predictor variables because if they're too correlated to the outcome variable, a model trained on both of these may just be predicting based on these two predictor variables. After consulting TAs, we decided to check the correlation between these two variables and the raw Spotify popularity score variable, `spotify_track_popularity`.

The correlation value for `weeks_on_chart` was 0.48. The correlation value for `peak_position` was -0.27. The magnitude of the correlations are not very high (less than 0.8, the threshold we discussed with TAs), so we proceed with caution and include these predictor variables. As mentioned before, we would like to justify that a song's Spotify popularity *score* is not an exact mirror of its Billboard popularity, as it's possible a song has high Spotify score yet lower Billboard popularity and vice versa. One metric is used for Spotify customer recommendation, the other to measure public sentiment.

Methodology

Our response variable is a binary categorical variable, hence we use logistic regression. We wish to determine which features of a song impact whether or not it `is_popular` according to Spotify, so we create and compare different models. Before we start, we check conditions for fitting a logistic model:

Linearity: for our quantitative predictors, we use `emplogitplot1` to check linearity between them and log-odds. At a glance, all variables except `spotify_track_duration_ms` seem to have a fairly linear relationship with log-odds. We drop `spotify_track_duration_ms` for this reason.



Randomness: in our case, we have access to all songs (the population of songs) from Spotify that have appeared on the Billboard Hot 100, thus this condition isn't relevant.

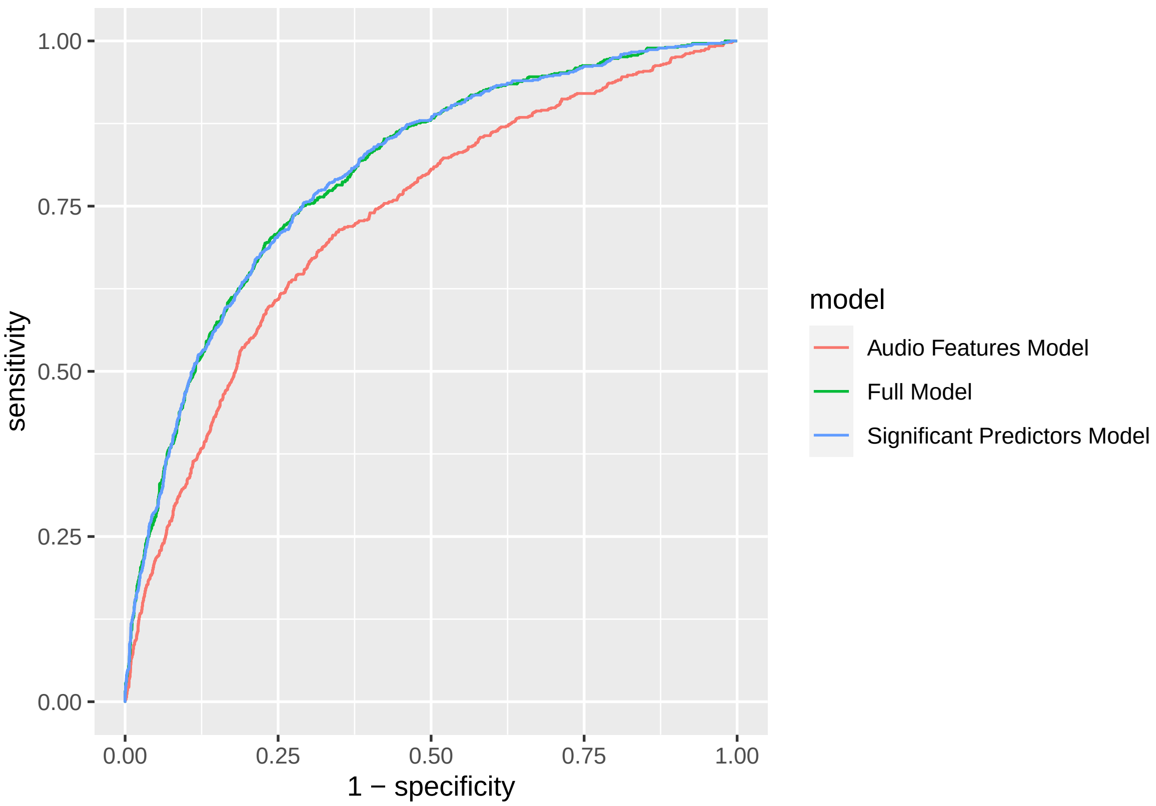
Independence: we can reasonably assume one song's features doesn't affect another song's features.

The first recipe we create will include all variables, our full model. (Variables include: `peak_position`, `weeks_on_chart`, `loudness`, `energy`, `key`, `mode`, `liveness`, `valence`, and `tempo`). Full model output can be found in the Appendix.

To create our second model, we will select the predictor variables with p-values below our significance level of .05 from our first full model. We created this model because a low p-value indicates that the coefficients of these predictors are significant in the regression, and therefore significant predictors. We'd like to note that we also remove the `key` predictor variable because only 1 out of 11 levels had a significant p-value. The predictors that are statistically significant based on p-value (and are therefore included in this model) are `peak_position`, `weeks_on_chart`, `loudness`, `liveness`, `valence`, `tempo`, and `mode.` Full model output can be found in the Appendix.

We then create a third model with only the Spotify audio features (nothing from Billboard Hot 100) to determine if popularity can still be predicted without some of the variables with more obvious correlation (such as `peak position` and `weeks_on_chart`). We did this because we'd like to see how a model without Billboard Chart data performs, since chart data is delayed and may take record labels longer to compile. The variables in this model are `loudness`, `energy`, `key`, `mode`, `liveness`, `valence`, and `tempo`. Full model output can be found in the Appendix.

To compare models, we will use ROC curves.



The ROC curves for our full model and the one including only the significant coefficients appear similar. The area under the curves for the full model and the model containing significant coefficients are 0.799 and 0.7993, respectively. The ROC curve of the audio model is indicative of a noticeably worse performance compared to the other two models. Its area under the curve is 0.73. Obeying the principle of parsimony, we opt for the smaller model containing just significant coefficients.

Results

Our final model is the one including only statistically significant predictor variables:

term	estimate	std.error	statistic	p.value
(Intercept)	1.667	0.199	8.397	0.000
peak_position	-0.018	0.001	-13.766	0.000
weeks_on_chart	0.087	0.003	25.944	0.000
loudness	0.214	0.009	24.002	0.000
liveness	-0.761	0.177	-4.308	0.000
valence	-1.601	0.114	-14.079	0.000

term	estimate	std.error	statistic	p.value
tempo	0.003	0.001	3.122	0.002
mode_X1	-0.321	0.058	-5.501	0.000

$$\log(\frac{\hat{\pi}}{1-\hat{\pi}}) = 1.67 - 0.02 \times peakPosition + 0.09 \times weeksOnChart + 0.21 \times loudness$$

$$-0.76 \times liveness - 1.6 \times valence + 0.003 \times tempo - 0.32 \times mode$$

Since we're aiming to predict songs that will be considered popular by Spotify, we interpret a few of the positive coefficients. That is, when we increase the value of variables associated with those coefficients, the log-odds of a song being popular are expected to increase. Artists who want to make popular songs by Spotify's metrics should increase the following characteristics of their songs:

For each additional decibel increase of loudness, the odds that a song is popular are expected to multiply by 1.24 on average, holding all other variables constant.

For each additional week on Billboard Hot 100 chart, the odds that a song is popular are expected to multiply by 2.39 on average, holding all other variables constant.

Part of our research question was prediction. The final model's confusion matrix on the testing dataset is

```
          Truth
Prediction   NO  YES
       NO  2289  519
       YES  176  311
```

The sensitivity of our predictor is $\frac{311}{311+519} \approx 0.375$. The specificity is $\frac{2289}{2289+176} \approx 0.929$.

Discussion & Conclusion

We set out 1) to find significant predictors of whether or not a song is popular and 2) to see if we could build an accurate model that predicted whether or not a song is popular.

To address 1: in conclusion, we discovered that predictors with greatest significance (in terms of p-value) include `peak_position`, `weeks_on_chart`, `loudness`, `liveness`, `valence`, `tempo`, and `mode`. This was determined by a hypothesis test in which the H_0 stated that the coefficients of these variables were 0, and their p-values ended up being significant enough to reject the null.

We solidified this claim by comparing the areas under the ROC curve of a model containing just these predictors (area of 0.8) with other models (areas of 0.8 and 0.73) to show that the model containing just significant coefficients performed on par or better. While we were able to determine significant predictors of song popularity, our initial hypothesis stating an increase in valence, energy, and peak position would be the significant predictors of a popular song wasn't entirely correct because the coefficient of `energy`'s hypothesis test (testing if its coefficient is nonzero) did not reject the null, and `valence`'s coefficient is negative. Our hypothesis for an increase in `peak_position` increasing the likelihood of being popular did match our model because it has a significant coefficient with value less than 0 (By higher the position, we mean lower position value since position 1 is the best position). Hypothesis aside, our results section showed how an artist who wants a high song popularity ought to increase the loudness of their songs.

To address 2: looking at our confusion matrix in the results section, our predictive model has misclassification rate

$$\frac{176 + 519}{176 + 519 + 2289 + 311} \approx 21.1\%$$

This isn't too good nor too bad for our purposes of giving record labels a general idea of if a song is popular. However, our model's sensitivity (calculated as 0.375 in the results section) isn't good. This means our model isn't great at predicting a song will be popular when it is actually popular. The misclassification rate for popular songs is 62.5%, while only 7% for unpopular songs. This could be due to the inbalanced nature of our dataset (which is to be expected since we made the threshold for popularity based on the 75th percentile).

Some limitations our project had is the fact that music and musical rankings are constantly changing — ideally, we should have access to the most up to date data that is constantly updating. However, this is obviously incredibly difficult in terms of data maintenance and exploration, so our project used a previous ranking. One way this could have affected our conclusions is that while we may technically drawn the correct conclusion from our dataset, it still begs the question on whether our data is representative of the actual musical sphere in the modern world. This question is larger in scope than our original research question, and if our proposal had included that question, or if anyone in the future tries to use our data to deduce the answer to this question, it may be better to start anew and use more updated musical rankings, as people's tastes change over time and our conclusions may not hold in 5 to 10 years. Our hunch is Spotify keeps this data secret on purpose so as to not publicize how it derives song popularity. Furthermore, a limitation we had was that instead of predicting the exact popularity score (0-100), we predicted a binary variable indicating whether or not the song was in the top quartile of popularity scores. The choice of labelling the top quartile of popularity scores as popular is one that is not an exact science–we determined this threshold after consulting among ourselves and teaching assistants. While we justified our choice in our data description, it would probably be more helpful to record labels to be able to predict the exact numerical score. Lastly, we also realize that the songs we train and tested on needed to be popular enough to first make it into the Billboard charts and hence included in our dataset. Regarding this, we did note in the EDA that despite a song appearing on Billboard, it did not necessarily imply a high Spotify popularity (score), hence our analysis was still necessary.

In terms of future work, this project could definitely be expanded to include more musical ranking metrics than just the Billboard. There are obvious important predictors that have been left out of our model that we didn't have access to. For instance, song click rate, how many people have listened to the song in the past week, etc.. Additionally, an interesting idea would be to see if these trends we discovered in our analyses hold in other regions of the world that do not primarily follow the Billboard rankings — an interesting project could be done there to determine if music taste is universal across multiple cultures in terms of the same factors such as valence, energy, etc. Another potential project would be to contrast an algorithm's predicted value for a song's popularity versus asking real people to rate a song on its certain features, then extracting the true popularity value from it.

Appendix

Full Model Output:

term	estimate	std.error	statistic	p.value
(Intercept)	1.612	0.263	6.126	0.000
peak_position	-0.018	0.001	-13.624	0.000
weeks_on_chart	0.086	0.003	25.873	0.000
loudness	0.212	0.012	17.513	0.000
energy	-0.011	0.211	-0.054	0.957
liveness	-0.719	0.179	-4.021	0.000
valence	-1.598	0.127	-12.614	0.000
tempo	0.003	0.001	2.989	0.003
key_X1	0.251	0.116	2.169	0.030
key_X2	0.061	0.116	0.523	0.601
key_X3	-0.115	0.168	-0.687	0.492
key_X4	0.075	0.122	0.617	0.537
key_X5	-0.187	0.122	-1.531	0.126
key_X6	0.204	0.129	1.577	0.115
key_X7	-0.146	0.112	-1.305	0.192
key_X8	0.109	0.130	0.841	0.401
key_X9	0.031	0.113	0.274	0.784
key_X10	-0.161	0.129	-1.250	0.211
key_X11	0.204	0.123	1.656	0.098
mode_X1	-0.301	0.060	-4.974	0.000

Significant Predictors Model Output:

term	estimate	std.error	statistic	p.value
(Intercept)	1.667	0.199	8.397	0.000
peak_position	-0.018	0.001	-13.766	0.000
weeks_on_chart	0.087	0.003	25.944	0.000
loudness	0.214	0.009	24.002	0.000
liveness	-0.761	0.177	-4.308	0.000
valence	-1.601	0.114	-14.079	0.000
tempo	0.003	0.001	3.122	0.002
mode_X1	-0.321	0.058	-5.501	0.000

Audio Model Output:

term	estimate	std.error	statistic	p.value
(Intercept)	2.268	0.224	10.149	0.000
loudness	0.249	0.012	21.618	0.000
energy	-0.135	0.196	-0.690	0.490
liveness	-1.026	0.169	-6.062	0.000
valence	-1.813	0.118	-15.308	0.000
tempo	0.002	0.001	2.475	0.013
key_X1	0.364	0.107	3.397	0.001
key_X2	0.000	0.108	-0.003	0.998
key_X3	-0.093	0.157	-0.592	0.554
key_X4	0.105	0.114	0.927	0.354
key_X5	-0.128	0.114	-1.128	0.260
key_X6	0.244	0.121	2.019	0.043
key_X7	-0.092	0.105	-0.884	0.377
key_X8	0.169	0.120	1.401	0.161
key_X9	-0.003	0.106	-0.033	0.974
key_X10	-0.154	0.120	-1.278	0.201
key_X11	0.253	0.115	2.206	0.027
mode_X1	-0.336	0.056	-5.950	0.000