What makes a song popular in 2020?

Analysing how the Mental Health and Economic crisis during 2020 have influenced music listening trends and how similiar this impact was to that of the Great Recession in 2008

Ananya Jha

21/12/2020

Abstract

Measures like lockdown and quarantine to combat COVID 19 might have led to an influx in the consumption of music and have consequently influenced popularity trends. This study aims to analyse and demonstrate which features of songs (if any) contribute to their popularity based on the current music listening trends. Stastical methods like Logistic Regression and Data Visualisation have been implemented to examine the similarity of these trends during The Great Recession of 2008 and the Pandemic in 2020. This analysis investigates whether the global environment has an impact on the music listening trends among the public. The results show substantial similarity between the listening trends and people's preferences of music in the two periods and testify the hypothesis that music preferences are, in fact, impacted significantly in times of crisis.

Code and data supporting this analysis is avaliable at: https://github.com/jhanan1/What-makes-a-song-popular-in-2020-

Keywords

Data Analysis, Music, Logistic Regression, Data Visualization, COVID19, The Great Recession, SpotifyAPI

Introduction

The COVID 19 pandemic has led to the creation of a new reality filled with challenges to mental health because of unemployment and economic loss, health concerns, and the loneliness induced by quarantine. Lockdowns across the globe have had people partake in numerous self-indulgent activities and one of them is listening to music. Songs like "Stuck with You" and "If the World was Ending" captured emotions that everyone was experiencing and garnered global popularity. For music enthusiasts such as myself, I thought it would be interesting to investigate the constituents that make a song popular in the current context and the relationship between crises and music trends. For the latter, comparison to another global crisis with similar consequences, for instance, The Great Recession of 2008 would assist the analysis.

Predicting the popularity of the songs based on their features can be done in many ways. I explored Linear Regression and Logistical Regression and finally decided to proceed with Logistic Regression as some of the variables didn't hold the assumptions of linearity required for Linear Regression. Logistic Regression is used when the response variable is binary to predict a "Yes/No" answer. In this case, if a song was popular it had a pop_log (binary response variable) value of 1, and 0 otherwise. The logistic model gives log-odds coefficient of the features of a song (i.e describes how much of an impact the song feature has on the probability of the song to be popular). For comparison to 2008 listenting trends, the predict function in R was used.

The 2020 model was constructed using a dataset obtained from Kaggle which contained over 140,000 songs. This dataset was based off of Spotify's collection of data and their popularity score rating. A popularity score with a cut off value of 60 was used, i.e. songs with a popularity score above 60 were considered popular and those below were considered not popular (as this classified top 11% of the songs as popular). Data for Billboard Top 100 songs of 2008 (Github) was procured from Spotify for developers API and this was further run against the model using the predict function to determine its popularity. More details about this are in the Methodology section.

Out of the 100 songs chosen from 2008, one song was not available on Spotify, therefore, the model only used 99 songs. 87 out of those 99 songs were predicted correctly by the model using the 0.5 cut off for probability. The songs that were below the 0.5 cutoff were also very close to the cutoff with values ranging from 0.4 to 0.49. The trends observed in 2020 and 2008 were also very similar just as expected. More details about this have been presented in the Result of analysis and in the Conclusion section of the report. These are helpful in many ways including helping musicians create music that people prefer listening to during times of crisis and consequently, making more revenue by creating popular music.

Methodology

Data

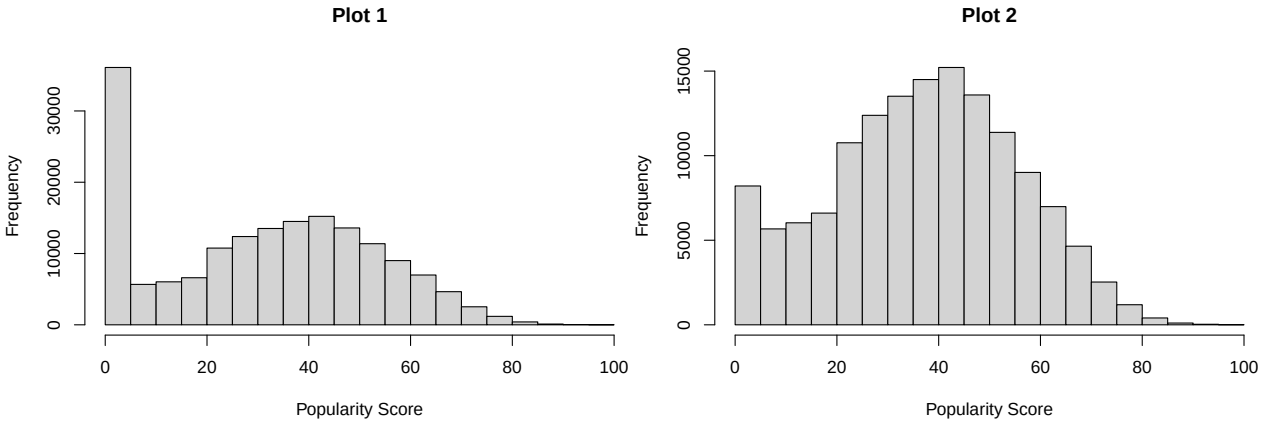
Data Sources

This project required the use of three different datasets obtained from various sources such as Kaggle, Github and Spotify API (using the spotifyR package). Through Kaggle, I acquired the 2020 popular songs dataset which contained over 140,000 songs grouped by artist, year, genre, and about 21 variables describing the features of these songs. However, after examining the large-scale data, I decided to use only six variables for my project (details available in the Data Description section). A summary of the unused variables and why they weren't used is presented in the Appendix. This data was largely numerical, and each song was attached with a unique ID; therefore, it was very useful for my regression analysis. Furthermore, the data had no missing values. The popularity score is using an algorithm that rates a song based on the total number of plays and how recent the plays were.

I used Github to acquire a list of Billboards Top 100 songs from 2008 and the songs were further used to create a playlist on my personal Spotify account. In order to analyse the features of the songs, I manually entered 99 songs (one was not available on Spotify) into the Spotify playlist and finally, through the use of Spotify API on R, data on the same features (the ones used in my model) of the top 2008 songs was obtained. In this analysis, my population is represented by the collection of all songs with their popularity score (on the same scale), sample is the set of songs that are present in my original dataset (obtained from Kaggle), and frame is the collection of songs on Spotify (the Spotify database).

Data Cleaning and Visualization

The first step to clean my data was checking the distribution of the response variable (popularity score). I detected that a lot of the songs on the playlist had a popularity score of zero which could cause a skewed analysis, therefore, I decided to remove all the observations with popularity score 0. After the data cleaning, my response variable followed a fairly normal distribution. Plots 1 and 2 illustrate the distribution of my response variable before and after filtering the songs with 0 popularity score.

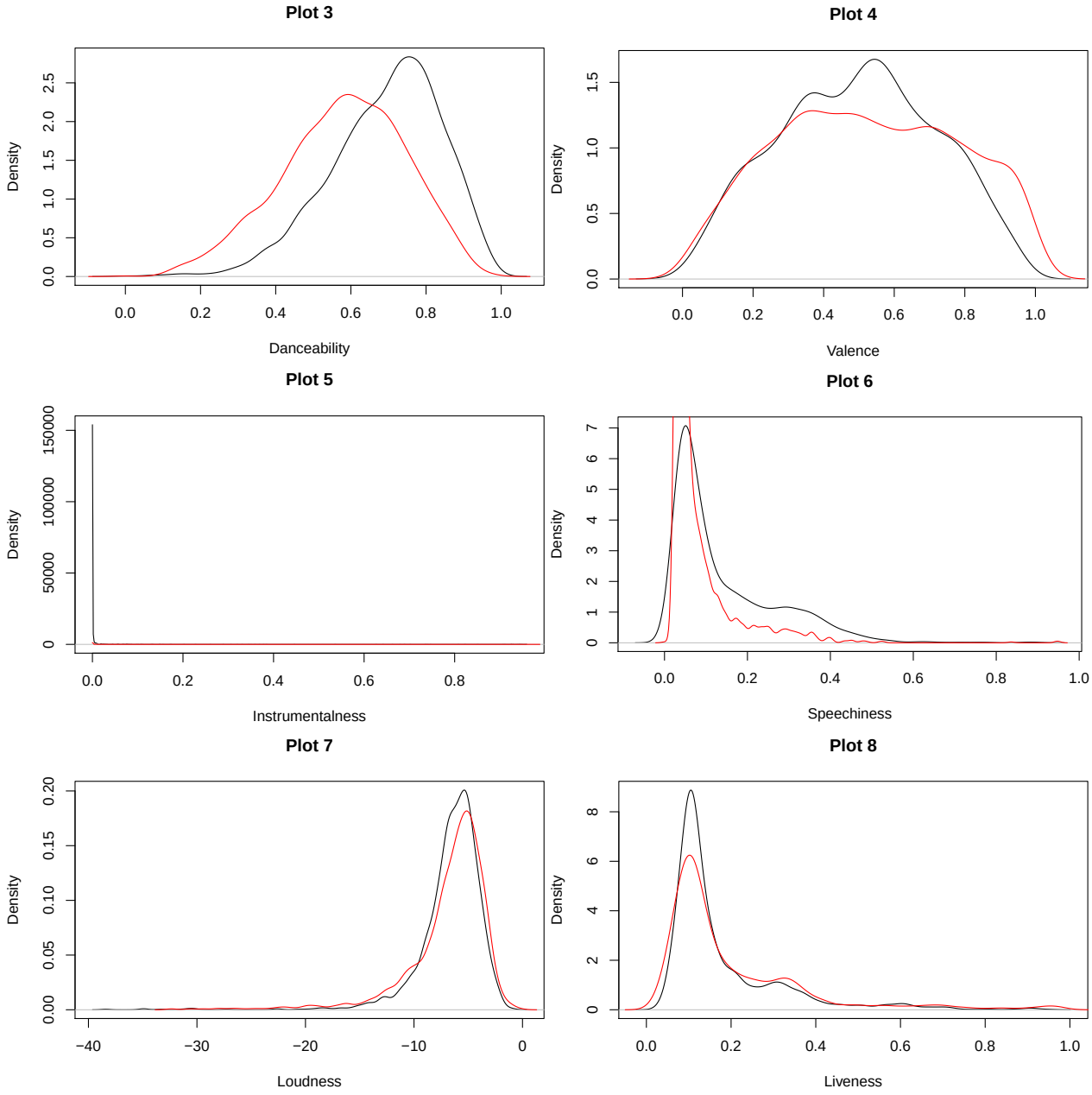


I also decided to change the year column (depicting the year in which the song was released) to years_since_released which involved subtracting the release year from 2020. Although this didn't really change my results, I felt years since a song was released as a variable would be more indicative and appropriate.

Furthermore, I had to create a new binary column (pop_log) assigning 1 to the songs with a popularity score greater than 60 and 0 to others. More details on how and why this cutoff was selected are present in the Methodology section.

Data Visualization

To visualize my data and gain an approximate understanding of similarity in songs released over these two years, I decided to plot the distribution of my variables (filtered depending on the year that the songs were released in). As can be seen in plots 3 through 8, overall trends were fairly similar with features like Loudness, Acousticness and Valence having the maximum similarity. This means that similar kind of songs were released in these two years. The plots are overlayed to demostrate relative distributions, red lines represent the distribution in 2020 and black represents 2008. I decided to keep only the plots of final variables and the others can be found in the Appendix.



Relevant Data Description

Table 1 displays my main variables of interest and their description as obtained from the Spotify for Developers website. More information about these can be found in the Appendix or on the website.

Table 1: Table describing all variables

Variable Name	Description
Year since relased (int)	Representing the number of years since the release of the song
Valence (Float)	"A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track."
Intrumentalness (Float)	"Predicts whether a track contains no vocals."
Speechiness (Float)	"Speechiness detects the presence of spoken words in a track. "
Danceability (Float)	"Danceability describes how suitable a track is for dancing ."
Loudness (Float)	"The overall loudness of a track in decibels (dB). "
Liveness (Float)	"Detects the presence of an audience in the recording. "
Tempo (Float)	"The overall estimated tempo of a track in beats per minute (BPM). "
Energy (Float)	"Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. "
Acousticness(Float)	"A confidence measure from 0.0 to 1.0 of whether the track is acoustic. "

The variable Years_since_release was interesting to include to understand whether time has a positive or negative impact on the popularity of a song. As likely as it is that a newly released song would be more "in" and preferred, I would also expect classics like Bohemian Rhapsody to remain perennially popular. I expected features like Valence, Danceability, Energy and Tempo to have a positive affect during the pandemic as positive music is uplifting and considered to be a mood booster during such low times. Intrumentalness and Speechiness both would state that people prefer songs with more words as they can sing along to the lyrics and gain moments of distraction from the outside world. Loudness is another factor that I wanted to consider because various studies have reported that extreme and loud music can help people deal with anger, an emotion that is prevalent presently. Since the pandemic caused a cancellation of all concerts, I also assumed that to make up for this, people would prefer listening to music with more live audience, so I wanted to include Liveness as well. Acousticness could go either way and was mainly included because I was curious to find out if it really affects the popularity of a song as it totally depends on personal music taste.

Model and Methodology

The main software that I used for my analysis was R. Initially, I wanted to construct a model to predict the popularity score of a song, so I tried Linear Regression. However, my data distribution violated some assumptions and prerequisites (can be found in the appendix) and so I could not proceed with it. However, this procedure was useful to check which features were significant and to make a preliminary analysis.

My alternative plan was to make a Logistic Regression model which would predict whether a song is popular or not based on its features. I had to do some rigorous analysis to decide the cut off as there was plenty of undersampling, i.e. only 0.3% of songs had a popularity score more than 80. I noticed that as the cut off grew, my sample got more skewed and undersampling had an effect on the p values of my variables. Undersampling also causes a decrease in predicted probablity (range reduces) and so, after some calculations I decided on the cutoff being 60, which marked top 11% of the songs (which seemed like an appropriate criteria to clssify as popular). Therefore, I created a new binary variable called pop_log which had a value of 1 for popular songs (songs with a popularity score over 60) and 0 for songs that were not popular.

Using the new variable "pop_log", I constructed multiple Logistic Regression models with my variables of interest (that I thought would be relevant). I decided to use a few variables that I expected to be relevant (like years since released) and the rest of the variables that I wanted to investigate trends about. Some variables that I tried in my model that were surprisingly insignificant were Liveness and Tempo. Energy was slightly significant but caused efficiency to go down as it reduced significance of other variables, so I decided to not include that to avoid overfitting. Even though Loudness was least significant, it was still enough to keep in my model as I thought it would be one of the features of music that might be different for

this year (studies have shown extreme music helps people be calm). All my other variables were significant enough to be kept and this model had the least AIC score compared to all other models that I tried.

The Correlation Table below (Table 3) shows the correlation coefficient between my variables that were used in the final model. None of my variables have a high correlation and so its safe to proceed with these. In Plot 9 we can also see the ROC curve for the final model. I have attached an ROC curve for 85 as a cutoff in the Appendix (to show comparison between the two cutoffs) but after trying multiple cutoffs, I finally picked the 60 cutoff model. The summary of my model also showed that Fisher's Scoring Algorithm needed seven iterations to perform the fit so the model did indeed converge, and had no trouble doing it[1].

Table 2: Correlation Table

	years_since_released	valence	instrumentalness	speechiness	danceability	loudness	liveness
years_since_released	1.00	0.04	0.19	0.00	-0.20	-0.45	0.04
valence	0.04	1.00	-0.22	0.02	0.55	0.30	0.00
instrumentalness	0.19	-0.22	1.00	-0.11	-0.27	-0.42	-0.04
speechiness	0.00	0.02	-0.11	1.00	0.20	-0.01	0.15
danceability	-0.20	0.55	-0.27	0.20	1.00	0.29	-0.11
loudness	-0.45	0.30	-0.42	-0.01	0.29	1.00	0.07
liveness	0.04	0.00	-0.04	0.15	-0.11	0.07	1.00

After careful consideration and looking at all the above mentioned criteria, my final equation (with coefficient values) for the model was:

$$\log\left(\frac{p(\text{pop\_log}=1)}{1-p(\text{pop\_log}=1)}\right) = -0.2-0.1(\text{YR})-0.38(\text{VL})-0.77(\text{IS})+0.01(\text{LO})-0.48(\text{LI})-0.56(\text{SP})+1.29(\text{DB})+\epsilon$$

Where p(pop_log=1)=p(x) which is the probablity that the song has pop_log value of 1, i.e the probablity that it will be popular and the independent variables are represented by:

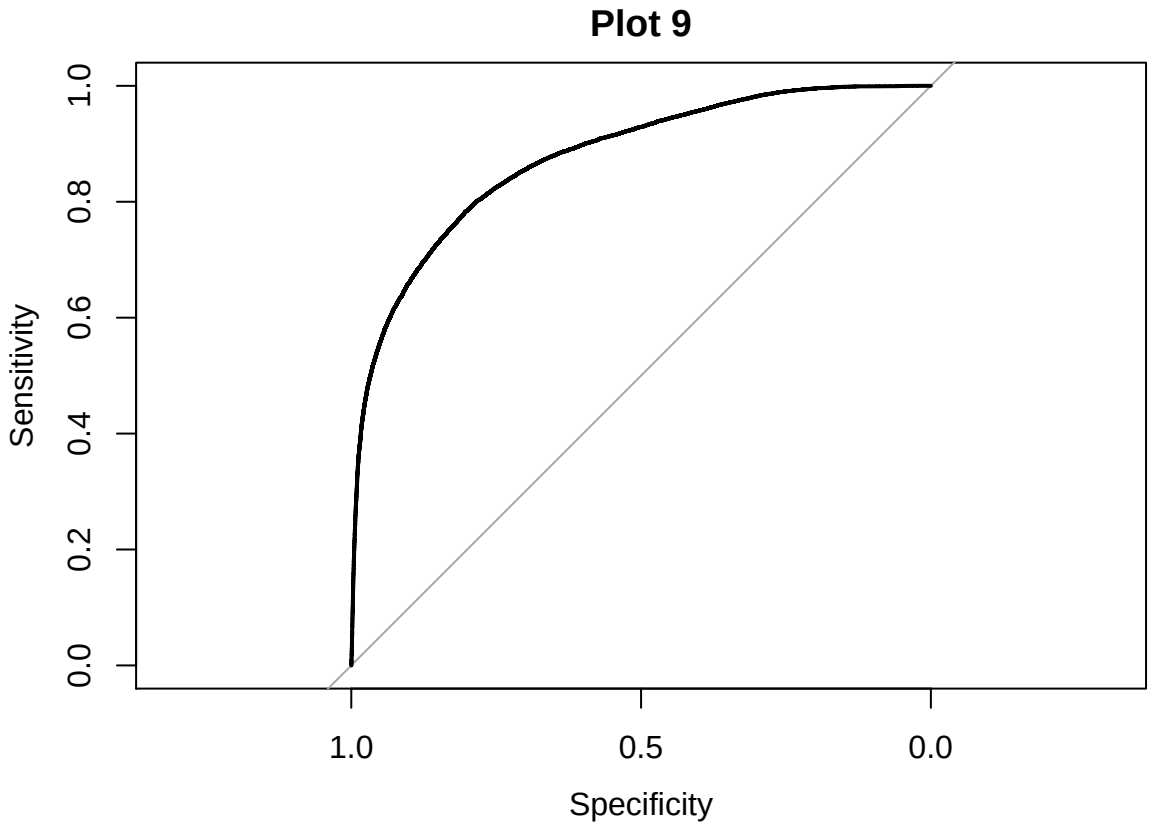
Abbreviation	Variable it represents
YR	Years since the song was released
VL	Valence
IS	Instrumentalness
LO	Loudness
LI	Liveness
SP	Speechability
DB	Danceability

Each estimated coefficient is the expected change in the log odds of a song being popular for a unit increase in the corresponding predictor variable holding the other predictor variables constant at certain value. Each exponentiated coefficient is the ratio of two odds, or the change in odds in the multiplicative scale for a unit increase in the corresponding predictor variable holding other variables at certain value. More analysis on the coefficients and the affects of different variables in predicted probablity is in the results section[1].

Now that I had a model representing the music listening trends in 2020, all that was left was to find a way to compare these trends to those in 2008. My initial idea was to get popularity data from 2008 and construct a model with similar variables and compare the coefficients of the two models. However, I couldn't find any details about popularity of songs (or data) from 2008, I instead tried another approach. I obtained a dataset with top 100 most popular songs (according to Billboard rating). Billboard rating is one of the most reputed

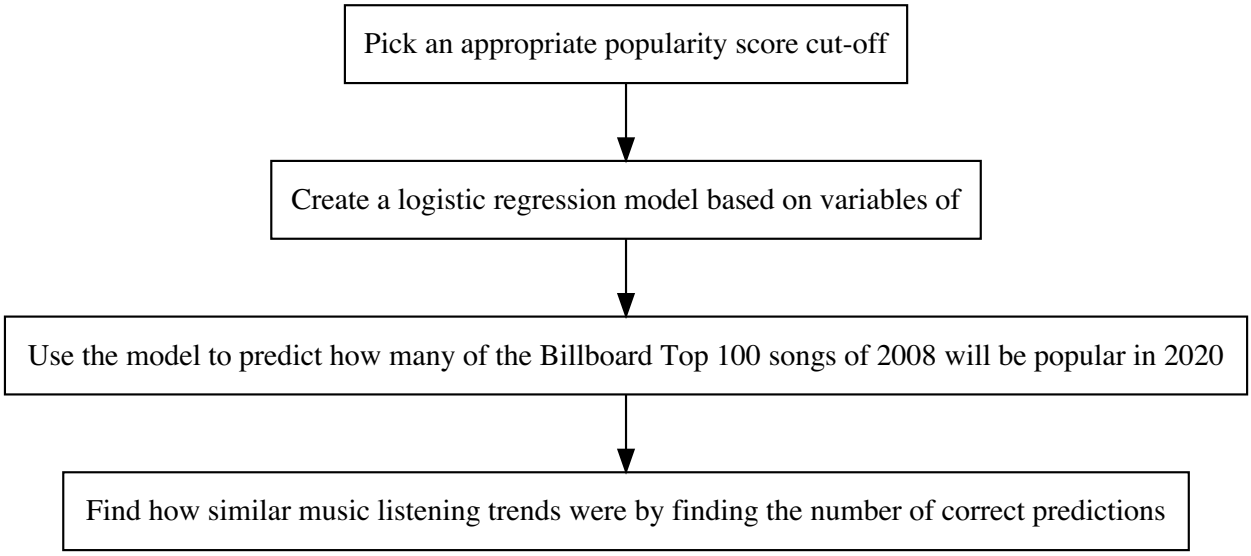
music ratings in the industry. Their Top 100 year-end tracks are decided through 3 criteria- Radio Airplay, Sales Data, and Streaming Data. Then I obtained all the variable details for 99/100 of these songs (one song was not available) through the Spotify API on R.

Finally, using the predict function in R, I calculated how many of these songs will be popular according to my model (which was based on the current music trends). I used predicted probably more than 0.5 to assign pop_log of 1 meaning song would be popular according to the model. I finally calculated the sum of the songs that would be popular according to my model and this would explain the extent of similar music listening trends in 2008 and 2020.



Results

Summary of steps invloved in the analysis



Final Model specifications:

Table 4 displays the coefficient (log-odds ratio) of a song being popular based on its value for a variable provided other variables are fixed constant. A song with a high Danceability and Loudness score are more likely to be popular, whereas songs with higher features like Liveness, Speechiness, Instrumentallness, Valence and Years_since_release are less likely to be popular. This was surprising since I expected Valence to have a high positive impact in 2020.

Table 4: Coefficient table

names	x
(Intercept)	-0.2020745
years_since_released	-0.0955227
valence	-0.3794074
instrumentalness	-0.7724788
loudness	0.0061514
liveness	-0.4823087
speechiness	-0.5639188
danceability	1.2928893

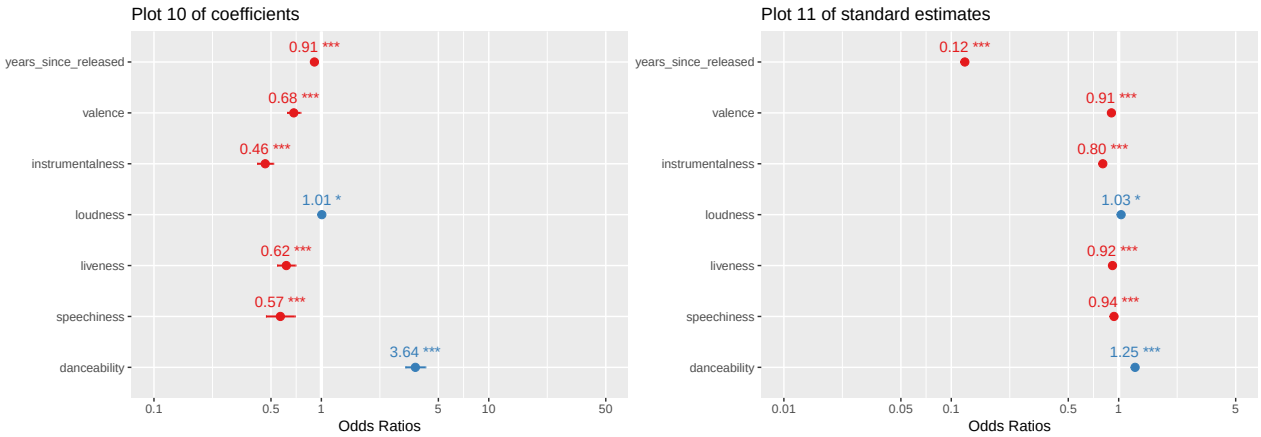
Although the variables seem to have a very slight effect on the predicted probablity, all variables are extremely signifcant (at the 0.05% significance level) and have a very small p-value [1] (as can be seen in Table 5) . The model also has a small pseudo R^2 value of 0.38(R^2 is generally found in only OLS regression and here it is estimated so called Pseudo) and AIC score 69346 (which was least compared to all other tried models).
Overall in the summary table (Table 5), it looks like the most significant factor in deciding the popularity of a song is Danceability, and understandably so. Listening to danceable music builds up morale and boosts energy which is extremely important in these times of crisis. A slightly less significant feature is Loudness and it has a positive effect on the predicted probability which also holds according to my research that loud and extreme music helps become calmer[4]. Instrumentalness, Years_since_released, Valence, Liveness and Speechiness are important features as well, but with negative magnitudes, indicating that, all else equal, a one-unit increase in either of those will result in a decrease in the odds of a song being popular[2]. This was a

little surprising since I estimated a song having some live audience element would remind people of concert days and provide them some sort of "virtual concert experience".

Table 5: Summary table of the model

term	estimate	std.error	statistic	p.value
(Intercept)	-0.2020745	0.0489634	-4.127054	0.0000367
years_since_released	-0.0955227	0.0009129	-104.634023	0.0000000
valence	-0.3794074	0.0465940	-8.142833	0.0000000
instrumentalness	-0.7724788	0.0563323	-13.712896	0.0000000
loudness	0.0061514	0.0026573	2.314943	0.0206160
liveness	-0.4823087	0.0642873	-7.502394	0.0000000
speechiness	-0.5639188	0.1012550	-5.569293	0.0000000
danceability	1.2928893	0.0700218	18.464106	0.0000000

Along with a coefficient table, I have attached Plots 10 and 11 to visualize the plot of coefficients and of standard estimates respectively. The plot of coefficients is useful for determining the direction of increase and that of exponentiated magnitudes is useful for checking predicted probablities. The variables in red have a negative impact on predicted probablity and those in blue have a positive.



Out of 99 popular songs (from 2008) whose features were available on Spotify, my model predicted that **87** would be popular according to the current music listening trends. I assigned songs which had a higher probability of being popular than not popular a pop_log value of 1 and summed up all those values. This means, 87.87% of these songs will be popular if they were released right now, in 2020. It is safe to say, with a 87.87% success rate, the similarity between listening trends in these two years was significant. The songs that were not predicted correctly and their predicted proabablities are displayed in table 6 below. It is interesting to see that most songs had a probablity extremely close to 0.5(the cut-off) with an exception of "The Way I Are" (having 0.38).

Table 6: Predicted probablities of songs predicted to be Unpopular in 2020

Name	Probablity
Sexy Can I	0.4792449
Shake It	0.4746903
Paper Planes	0.4060253
Live Your Life	0.4588803
Say	0.4835654
Superstar (feat. Matthew Santos)	0.4623542
Get Like Me	0.4816482
Good Life	0.4099995
The Boss	0.4347736
In Love With a Girl	0.4677118
Feels Like Tonight	0.4991281
The Way I Are	0.3838931

Discussion

Summary

Through my analysis I investigated how similar the listening trends were in 2008 to that in 2020. My Null hypothesis was that the similarity will be a lot considering the amount of aggravented mental and financial stress that the world has face in these two years. Using logistic regression, I created model that would predict whether a song would be popular or not based on its features like danceability, valence etc depending on the current music trends(in 2020). The model was suprisingly significant considering music is supposed to be a very subjective topic. Out of all features, the danceabiltiy score of a song is expected to affect it's popularity the most. After obtaining the Billboard Top 100 data of the most popular songs from 2008, I used my model to predict how many of these songs would be popular today. The results were substantiated my expectation and it was obvious that people preferred listening to similar music in these two years.

Conclusion

Even though COVID19 has provided us an opportunity to rekindle our relationship with family, it has also introduced us to a great deal of loss. With increasing constraints from the government causing loneliness, music presented a pathway to be grounded and provided moments away from the hardships. My report aimed to investigate the relationship between the features of songs and how they impacted their popularity in the current situation (the dataset was updated 24 days ago).

My model found 7 extremely significant variables namely- years since the song was released, valence, instrumentalness, loudness, liveness, speechiness and danceability. Through examination of ROC curve, AIC score and p-values, my model proved to be appropriate. The final analysis and report concluded that 87.87% of the Billborad Top 100 songs from 2008 would have been popular if released today. The other 12 songs had a predicted probablity of over 0.4% (hence extremely close) of being popular. Therefore, the model predicted that the music trends and taste during 2008 (The Great Recession) and 2020 (The Pandemic) are extremely similar.

This holds with my intial hypothesis that since the impact of these two events were so extreme and similar to a great extent- including causing a massive mental health and economic crisis, it was bound to influence the trends in preferred music. It is really interesting to see this high amount of similarity and to think that even though so much has changed in the past 12 years, people's music preferences remain the same.

Results from my analysis can be beneficial for musicians to help them create pandemic appropriate music that helps people deal with their mental health. Moreover, as the popularity of a song is directly related to

the revenue it generates, this model can also be used by the music industry to increase their profits. Hence, the advantages of this analysis are twofold - it can be used to combat mental health issues and the economic crisis.

Weaknessess

One of the major roadblocks that I encountered while carrying out my study was that the distribution of the data was highly unbalanced, meaning that only 0.3% of songs in the dataset had a popularity score of over 80. In order to fix this I had to bring the cut off (for a song to be classified as popular) down to 60 as that consisted of top 11% of all songs in my dataset and seemed optimal. Additionally, an unbalanced data heavily affects the predicted probablity in logistic regression so bringing the cut-off down seemed appropriate. Also, a lot of the songs had a popularity score of zero (which could have significantly skewed my analysis) therefore, I decided to remove these songs while cleaning the data in order to have a more normal distribution.

The datasets used were obtained from varying sites, using different approaches and involved some hard coding which meant it was prone to human error. Furthermore, out of the Top 100 Songs from 2008(according to Billboard), one song (and its features) was not available on Spotify (however, this did not affect my analysis much).

Song popularity may also be strongly influenced by other factors that we failed to consider or those that weren't avaialable to us. Moreover, we have seen instances when songs unlikely to be popular have gained immense popularity. Music is known to be a very subjective topic and different people have different tastes and preferences.

Finally, I must acknowledge that through the course of my analysis, I have not looked at the data for the years between 2008 to 2020 (or before that) so it may be possible that the popularity trend for music remained consistent over the years. All we know is that the music listening trends were similar in 2008 and 2020 (and so were the song releases). If we used data from previous years to compare to 2020 trends (obtained through my model) instead of just 2008, my model may still predict most songs correctly. However, we must acknowledge that 87.87% is a good enough success rate to tell that music listening trends in the two years were very similar.

Next Steps

To make the model more significant trying polynomial transformations, or introducing higher order or interaction terms in the model might be a good idea. It would also increase the power of the model and the predicted probabilities in the case of Logistic Regression. However, while doing this, we still have to be cautious about not overfitting the model. Finding other explanatory variables by clustering the songs by genre, or the lyrics, or the number of followers of the artist as some of the parameters is also a good idea. However, determining the popularity of an artist and their followers would entail rigorous research and unavoidable hard coding.

Additionally, looking at songs that weren't as popular in 2008 and running that against our model as well, would solidify the analysis as it would also mean the model is good at predicting songs that are likely to not be popular. We could even do the same thing for data from other years (between 2008 and 2020) to see if the trends were consistent throughout those years. This could also help check whether 2008 had the maximum common popular elements predicted correctly, which would strengthen my initial claim.

Alternatively, if in the future, there is a dataset available with popularity score details (or just a binary variable indicating whether a song was popular or not) of all songs in 2008, we could use it to create another model involving the same variables and compare coefficients to get an actual idea of similarity in listening trends. All the above mentioned ideas could be combined in any combination as a next step in this research.

Lastly, another analysis could involve looking into the duration of time spent streaming music on Spotify as during the pandemic most people have been working from home and all concerts have been cancelled, so it would most likely produce a global impact. This would also have impacted the annual revenue of Spotify and might help to create a business model that instead of looking at just the popularity of the song, uses popularity score to predict the profit a particular song would incur irrespective of the year we are looking at. Moreover, as indicated in the weaknesses, I can analyze a change in popularity trends over the years and not just look at the trends in 2008 and 2020.

References

1) Lillis, D. (2020, January 16). Generalized Linear Models in R, Part 2: Understanding Model Fit in Logistic Regression Output. Retrieved December 22, 2020, from https://www.theanalysisfactor.com/r-glm-model-fit/

2) Devor, M. (2020). MattD82/Predicting-Spotify-Song-Popularity [Python]. https://github.com/MattD82/Predicting-Spotify-Song-Popularity (Original work published 2019)

3) Hall, S. (2020, May 27). This is how COVID-19 is affecting the music industry. World Economic Forum. https://www.weforum.org/agenda/2020/05/this-is-how-covid-19-is-affecting-the-music-industry/

4) Music, G. (2015, June 22). Listening to "extreme" music makes you calmer, not angrier, according to study. The Guardian. http://www.theguardian.com/music/2015/jun/22/listening-heavy-metal-punk-extreme-music-makes-you-calmer-not-angrier-study

5) Riederer, Y. X., Christophe Dervieux, Emily. (2020). 4.13 Convert models to equations | R Markdown Cookbook. https://bookdown.org/yihui/rmarkdown-cookbook/

6) Cohut, M. (2019, June 11). Human brains have evolved to "prefer" music and speech. https://www.medicalnewstoday.com/articles/325444

7) Ay, E. A. (2020, November 25). Spotify Dataset 1921-2020, 160k+ Tracks. https://kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks

8) Schaich, K. (2016, April 24). Kevinschaich/billboard. GitHub. https://github.com/kevinschaich/billboard

Appendix

More details about the variables present in the data:

1) Valence : 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).

2) Instrumentalness : "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.

3) Speechiness : The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

4) Danceability: It is based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

5) Loudness: Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.

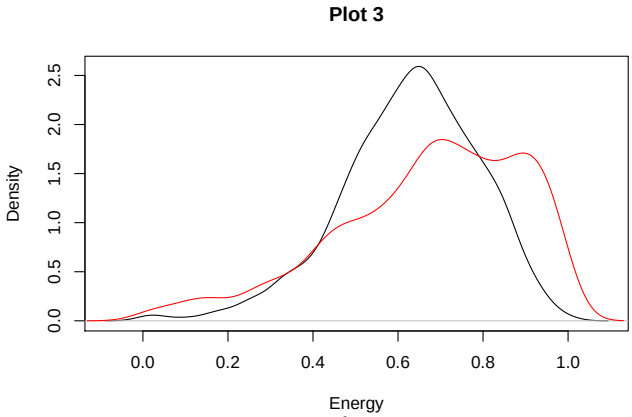
6) Liveness: Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

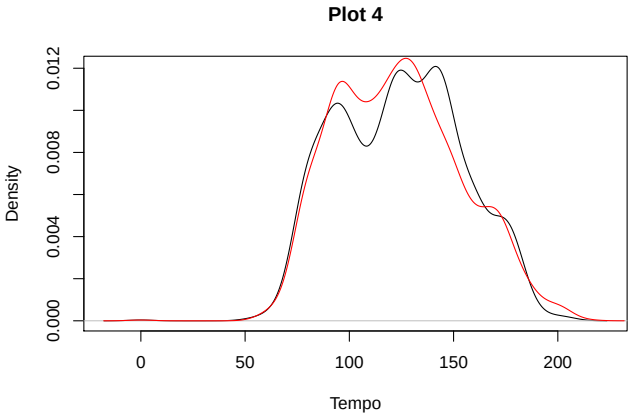
7) Tempo: In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

8) Energy: Typically, energetic tracks feel fast, loud, and noisy.

9) Acousticness: 1.0 represents high confidence the track is acoustic.

Distribution Plots for insignificant variables





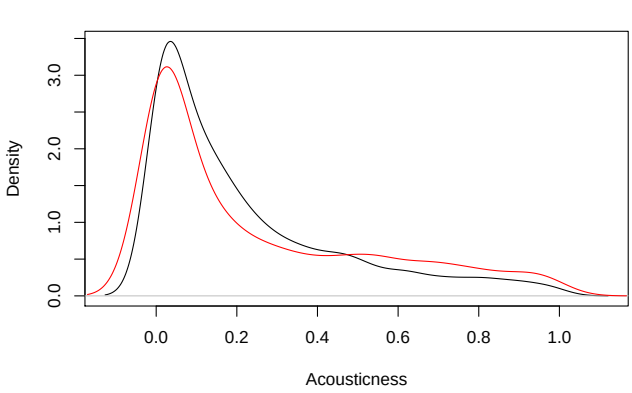
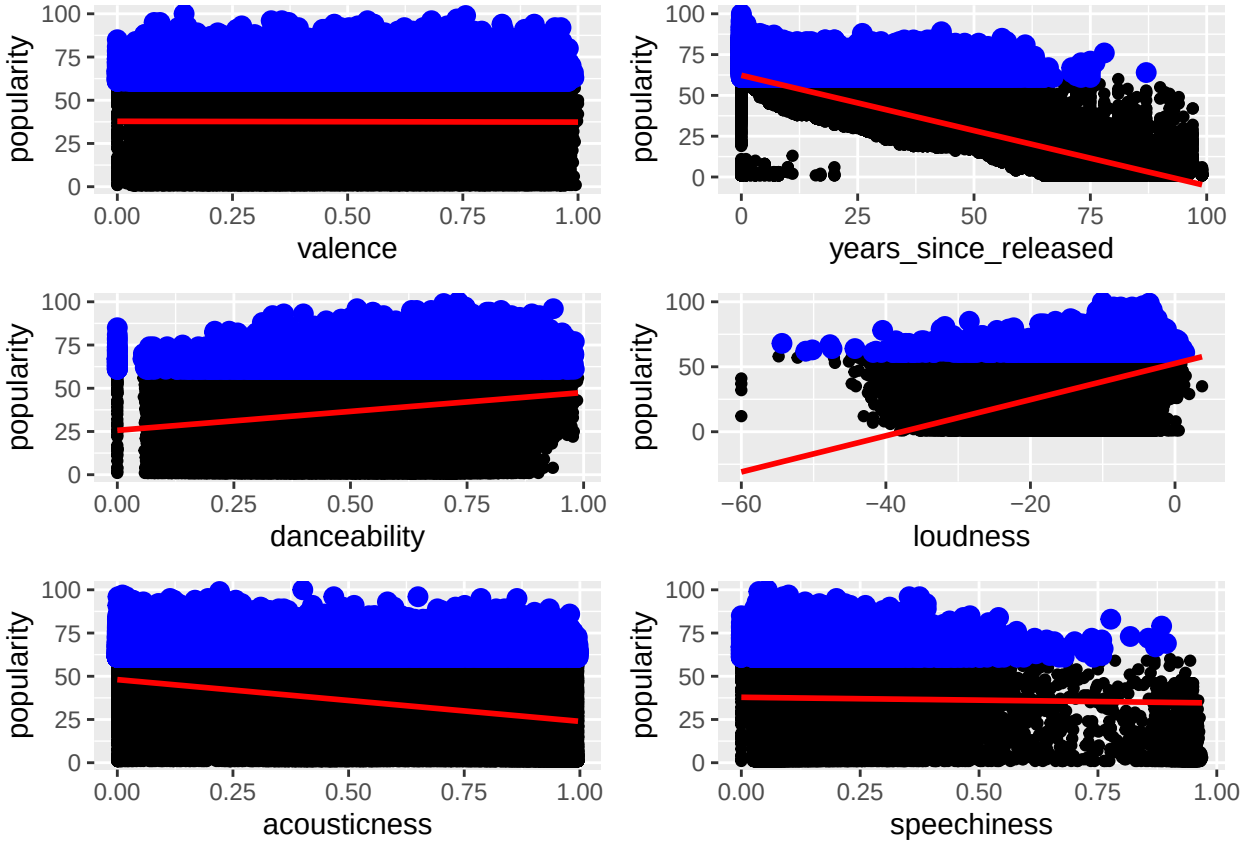


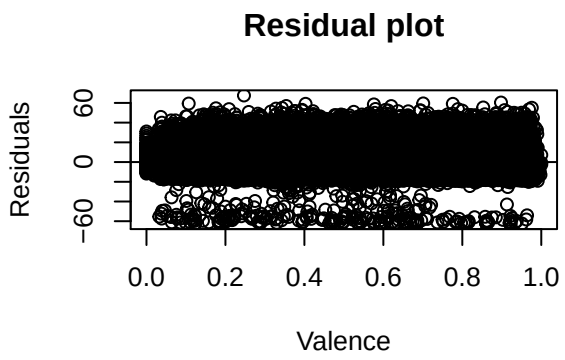
Table 7: Table x

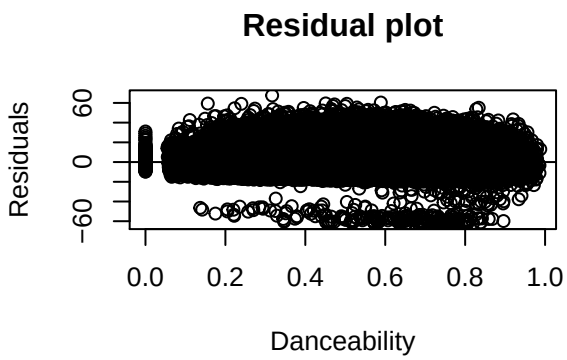
term	estimate	std.error	statistic	p.value
(Intercept)	62.0736099	0.1390902	446.283198	0
years_since_released	-0.6509786	0.0014906	-436.714220	0
valence	-0.7490638	0.1363655	-5.493059	0
instrumentalness	-2.9273119	0.1108088	-26.417673	0
loudness	0.1010334	0.0066162	15.270706	0
liveness	-3.0620988	0.1634553	-18.733560	0
speechiness	-4.9604569	0.2575443	-19.260593	0
danceability	3.9856941	0.2080024	19.161771	0

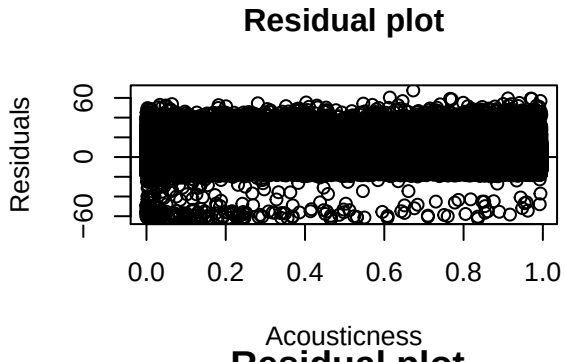
Preliminary analysis for a linear regression model:

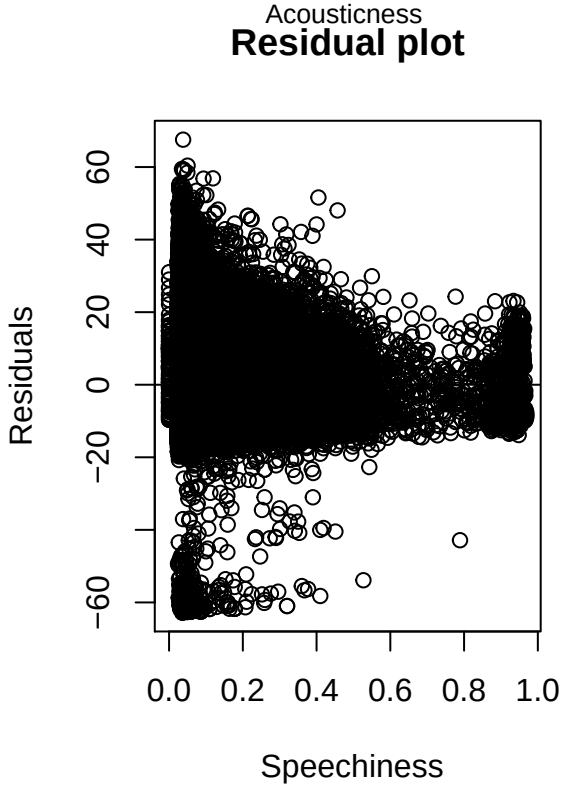


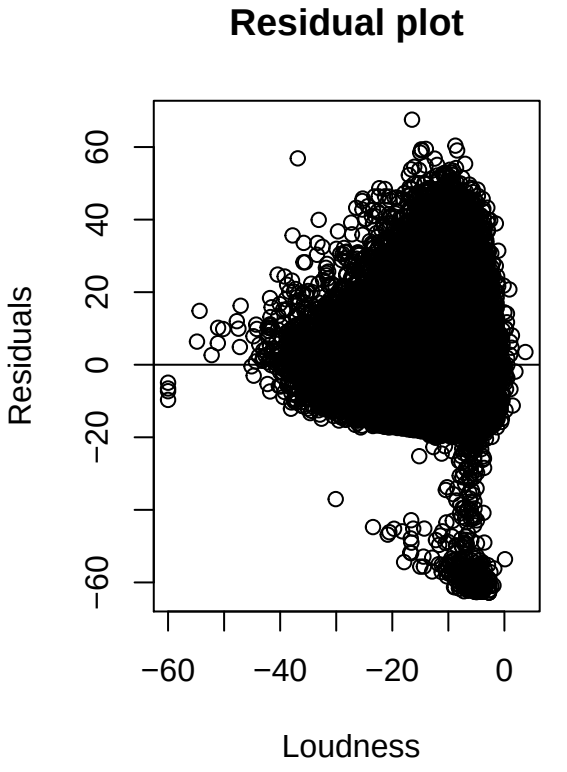
As we can see above in Figure X, none of our predictor variable have a significant Linear Relationship with popularity with valence,speechiness and acousticness having no linear relationship and lots of outliers. This violates one of the prerequisites for Linear Regression. Morever, in Figure X we clearly see residual plots are not randomly distrubuted or constant(with a visible megaphone effect in loudness and speechiness). So, even though a Linear Regression model had multiple pros- like a success rate of 96.96% (predicting 96/99 songs correctly with a popularity score of over 60 according to the cuttuog), a solid R^2 value, and greatly significant p values, proceeding with Linear Regression was not a good idea. The only way to do it would be by removing certain extrememly important variables from the model(which might also affect the model significance). Intuitively as well, it does not make sense that features like acousticness, danceability etc will have a linear relationship with popularity as music is not that objective. However, I wanted to investigate the success rate of a linear model so I have attached a summary report of my Linear regression model below.











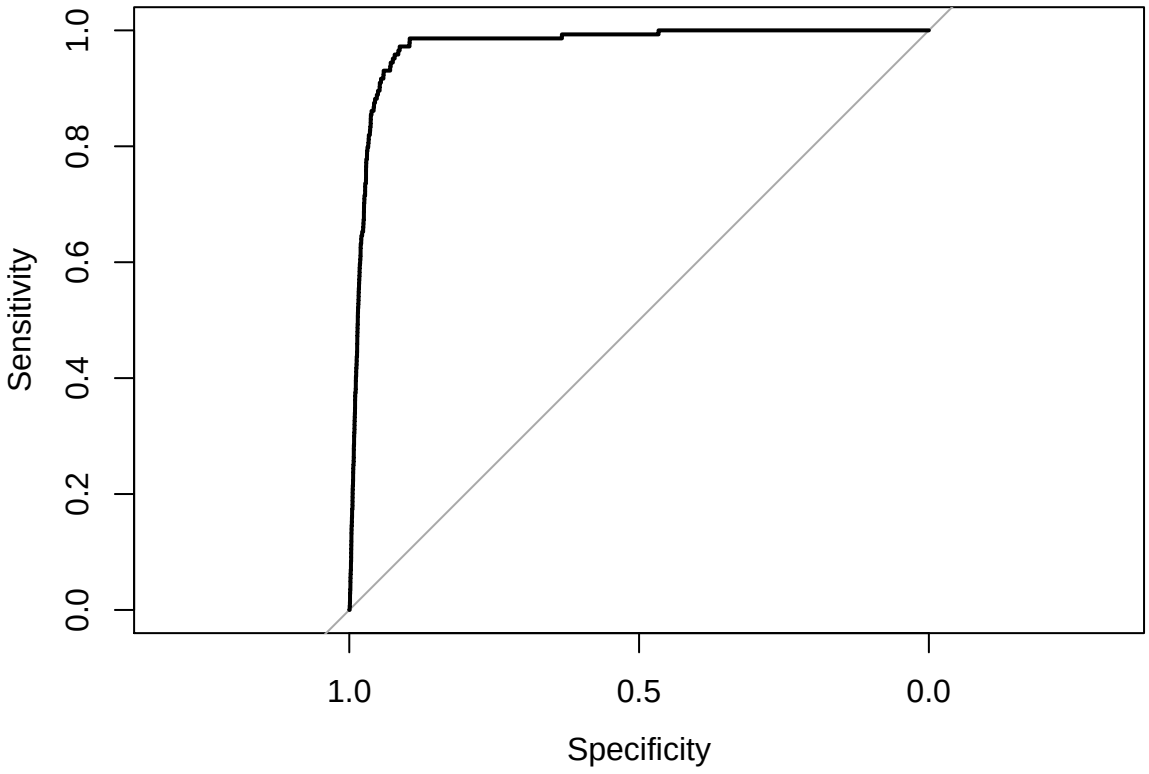
Approximate Linear Regression equation:

$$\text{popularity} = 62.07 - 0.65(\text{YR}) - 0.75(\text{VL}) - 2.93(\text{IL}) + 0.1(\text{LO}) - 3.06(\text{LI}) - 4.96(\text{SP}) + 3.99(\text{DB}) + \epsilon$$

with same coefficients as the logistic model.

Correct Predictions
96

ROC curve for 85 cutoff



All code involved in this analysis

```
knitr::opts_chunk$set(fig.pos = "!H", out.extra = "")
#devtools::install_github('charlie86/spotifyr')
#install.packages("dplyr")
#install.packages("ggplot2")
#install.packages("cowplot")
#install.packages("spotifyr")
#install.packages("plotly")
#install.packages("lubridate")
#install.packages("knitr")
#install.packages("equatiomatic")
#install.packages("jtools")
#install.packages("sjPlot")
#install.packages("sjlabelled")
#install.packages("sjmisc")
#install.packages("gridExtra")
#install.packages("pROC")
#install.packages("DiagrammeR)
#install.packages("gt")
#install.packages("glue")
library(dplyr)
library(ggplot2)
library(cowplot)
```

```r
library(spotifyr)
library(plotly)
library(lubridate)
library(knitr)
library(equatiomatic)
library(jtools)
library(sjPlot)
library(sjlabelled)
library(sjmisc)
library(gridExtra)
library(pROC)
library(DiagrammeR)
library(gt)
library(glue)
library(readxl)
library(kableExtra)


#devtools::install_github("haozhu233/kableExtra")
data_Raw <- read.csv("data.csv")
data_old <- read.csv("top.csv", fileEncoding = "Latin1") #since the file had an unrecognizable characte

#attach(data)
#attach(data_old)

#Removing songs with popularity = 0
data <- data_Raw %>% filter(popularity > 0)
#creating binary variable to specify if song is pop or not as of now
data<- data %>% mutate(pop_log = ifelse(data$popularity>60, 1, 0))
#years to years since relased
data<- data %>% mutate(years_since_released = 2020 - data$year)
#new dataset to plot popular songs
pop_data <- data %>% filter(pop_log > 0)

#density plots(plotting raw data)
P1<- hist(data_Raw$popularity, main = "Plot 1", xlab = "Popularity Score")
P2<-hist(data$popularity, main= "Plot 2", xlab = "Popularity Score")
#finding distribution of the variables I am interested in
dataTGR <- data %>% filter(year==2020)
dataCV <- data %>% filter(year==2008)
#comparison plots
plot(density(dataTGR$danceability), main = "Plot 3", xlab = "Danceability")
lines(density(dataCV$danceability), col="red")

plot(density(dataTGR$valence), main = "Plot 4", xlab = "Valence")
lines(density(dataCV$valence), col="red")

plot(density(dataTGR$instrumentalness), main = "Plot 5", xlab = "Instrumentalness")
lines(density(dataCV$instrumentalness), col="red")

plot(density(dataTGR$speechiness), main = "Plot 6", xlab = "Speechiness")
lines(density(dataCV$speechiness), col="red")
plot(density(dataTGR$loudness), main = "Plot 7", xlab = "Loudness")
```

```r
lines(density(dataCV$loudness), col="red")

plot(density(dataTGR$liveness), main = "Plot 8", xlab = "Liveness")
lines(density(dataCV$liveness), col="red")
df <- read_excel("Test.xlsx")
knitr::kable(df, caption = "Table describing all variables")%>% kable_styling(bootstrap_options = "borde
corr <- data %>% select(years_since_released , valence , instrumentalness , speechiness , danceability
res <- cor(corr)
knitr::kable(round(res, 2), caption = "Correlation Table") %>% kable_styling(bootstrap_options = "borde
#, include=FALSE}
#to find equation from the model
#equatiomatic::extract_eq(mainModel)
#finding equation with coefficients rounded
#equatiomatic::extract_eq(mainModel, use_coefs = TRUE)

#Our main logistic model
mainModel <- glm(pop_log~ years_since_released + valence + instrumentalness + loudness +liveness + spee
summary(mainModel)


prob=predict(mainModel,type=c("response"))
data$prob=prob
#ROC curve
g <- roc(pop_log ~ prob, data = data)
plot(g, main="Plot 9")
#spotify details, change id and secret code to your own code which can be found on spotify developers
id <- "5f8575efc9a44d0b9a407abb140315e2"
secret <- "edda7074c36148bfac64401432826fc0"
Sys.setenv(SPOTIFY_CLIENT_ID = id)
Sys.setenv(SPOTIFY_CLIENT_SECRET = secret)
access_token <- get_spotify_access_token()
#getting top 2008 songs features from playlist created; use as it is, no need to change anything
billboardTop<-get_playlist_audio_features("ttauspv1ije37agsszsaifvrj","0OPBjz8okoE9n01otiktjj",access_t
#extracting year from the date released column
billboardTop<- billboardTop %>% mutate(year =substr(billboardTop$track.album.release_date, 1, 4))

#creating the years since relased column
billboardTop<- billboardTop %>% mutate(years_since_released = 2008 - as.numeric(billboardTop$year))

#remove unecessary columns
billboardTop<- billboardTop %>% select(years_since_released,valence,track.name, danceability, speechines
#using predict function to predict probablity that a song will be popular
billboardTop$probablity <- mainModel %>% predict( billboardTop, type="response")
#assigning 1 to the columns with probablity of the song being popular > 0.5
billboardTop<- billboardTop %>% mutate(pop_log = ifelse(probablity > 0.5, 1, 0))
DiagrammeR::grViz("digraph {
  graph [layout = dot, rankdir = TB]

  node [shape = rectangle]
  rec1 [label = 'Pick an appropriate popularity score cut-off']
  rec2 [label = 'Create a logistic regression model based on variables of']
  rec3 [label =  'Use the model to predict how many of the Billboard Top 100 songs of 2008 will be popu
  rec4 [label = 'Find how similar music listening trends were by finding the number of correct predicti
```

```
  # edge definitions with the node IDs
  rec1 -> rec2 -> rec3 -> rec4
  }",
  height = 300)

t<-data.frame(broom::tidy(coefficients(mainModel)))
knitr::kable(t, caption = "Coefficient table") %>% kable_styling(latex_options = "hold_position")
knitr::kable(broom::tidy(mainModel) , caption = "Summary table of the model") %>% kable_styling(latex_o
# the percentage of popular songs according to my cut off
#print(sum(data$pop_log)/length(data$pop_log))

#Plotting log odds estimates
plot_model(mainModel, show.values = TRUE, value.offset = .3, title = "Plot 10 of coefficients")

#Plotting std estimates
plot_model(mainModel, type = "std",show.values = TRUE, value.offset = .3, title = "Plot 11 of standard

#getting the number of correct predictions
#print(correct_predictions <- sum(billboardTop$pop_log))

#t <- tibble(number_of_correct_predictions = 87)
#knitr::kable(t)

bp <- billboardTop %>% filter(pop_log == 0) %>% select(track.name, probablity)
bp <- rename(bp, Name=track.name, Probablity = probablity)
knitr::kable(bp, caption = "Predicted probablities of songs predicted to be Unpopular in 2020") %>% kabl
#comparison plots
plot(density(dataTGR$energy), main = "Plot 3", xlab = "Energy")
lines(density(dataCV$energy), col="red")

plot(density(dataTGR$tempo), main = "Plot 4", xlab = "Tempo")
lines(density(dataCV$tempo), col="red")

plot(density(dataTGR$acousticness), main = "Plot 4", xlab = "Acousticness")
lines(density(dataCV$acousticness), col="red")
sp1<-
  ggplot(data, aes(x = valence, y = popularity)) +
  geom_point() +
  geom_point(data=pop_data, aes(x=valence, y = popularity), colour = "blue", size = 3)+
   stat_smooth(method = "lm", col = "red")

sp2<-
  ggplot(data, aes(x = years_since_released, y = popularity)) +
  geom_point() +
  geom_point(data=pop_data, aes(x=years_since_released, y = popularity), colour = "blue", size = 3)+
  stat_smooth(method = "lm", col = "red")

sp3<-
  ggplot(data, aes(x = danceability, y = popularity)) +
  geom_point() +
  geom_point(data=pop_data, aes(x=danceability, y = popularity), colour = "blue", size = 3)+
  stat_smooth(method = "lm", col = "red")
```

```r
sp4<-
  ggplot(data, aes(x = loudness, y = popularity)) +
  geom_point() +
  geom_point(data=pop_data, aes(x=loudness, y = popularity), colour = "blue", size = 3)+
  stat_smooth(method = "lm", col = "red")

sp5<-
  ggplot(data, aes(x = acousticness, y = popularity)) +
  geom_point() +
  geom_point(data=pop_data, aes(x=acousticness, y = popularity), colour = "blue", size = 3)+
  stat_smooth(method = "lm", col = "red")

sp6<-
  ggplot(data, aes(x = speechiness, y = popularity)) +
  geom_point() +
  geom_point(data=pop_data, aes(x=speechiness, y = popularity), colour = "blue", size = 3)+
  stat_smooth(method = "lm", col = "red")


grid.arrange(sp1, sp2, sp3, sp4, sp5, sp6, nrow = 3)

prelModel <- lm(popularity ~ years_since_released + valence + instrumentalness + loudness +liveness + sp

knitr::kable(broom::tidy(prelModel), caption = "Table x")
#summary(prelModel)
#equatiomatic::extract_eq(prelModel, use_coefs = TRUE)

par(mfrow = c(2,2))
multi.res = resid(prelModel)
fitted=fitted(prelModel)
plot(data$valence , multi.res,
     ylab="Residuals", xlab="Valence",
     main="Residual plot")
abline(0, 0)

plot(data$danceability, multi.res,
     ylab="Residuals", xlab="Danceability",
     main="Residual plot")
abline(0, 0)
plot(data$acousticness, multi.res,
     ylab="Residuals", xlab="Acousticness",
     main="Residual plot")
abline(0, 0)

par(mfrow = c(1,2))

plot(data$speechiness, multi.res,
     ylab="Residuals", xlab=" Speechiness",
     main="Residual plot")
abline(0, 0)

plot(data$loudness, multi.res,
     ylab="Residuals", xlab=" Loudness",
```

```
     main="Residual plot")
abline(0, 0)


billboardTop$probablity <- prelModel %>% predict( billboardTop, type="response")
billboardTop<- billboardTop %>% mutate(pop_log = ifelse(probablity > 60, 1, 0))
#print(correct_predictions <- sum(billboardTop$pop_log))
kable(tibble("Correct Predictions" = 96))
#roc curve for cut-off of 85
data_next <- data %>% mutate(pop_log = if_else(popularity>85, 1, 0))
nextModel<- glm(pop_log~ years_since_released + valence+ acousticness + explicit + instrumentalness + sp
prob=predict(nextModel,type=c("response"))
data$prob=prob
library(pROC)
g <- roc(pop_log ~ prob, data = data_next)
plot(g)
```