**Spotify Trends**

Mengchen Gong

CSCI5502

mego6698@colorado.eduJuanfran Ferrer Peasley

CSCI4502

jufe2791@colorado.eduCarlos Prieto Fernandez

CSCI4502

capr5053@colorado.edu

**ABSTRACT**

There are more than 30 million songs in the Spotify library, which are from different artists, different genres and different moods. In our study we will be analyzing a dataset of 223 k songs and 18 of their properties. Our goal is to find what qualities are directly related to making a song popular, and in order to do this we will data mine the set and find correlations between the attributes that make a song popular. After we analyze our dataset and find what features are directly related to making a song popular, we will be able to look at other songs from our dataset, other datasets or even music charts and verify if our findings resemble what these datasets and music charts consider to be the most popular songs. If our findings resemble these from other datasets or charts we will be able to start drawing conclusions to what features make a song popular nowadays. The next step would involve us making predictions. What that means is that we will be able to look at the features from a song and estimate what chances that song has to become a hit according to our findings. The final step in our project will potentially be finding similar datasets that have the release date of the songs so we can analyse how the popularity of the songs changes over time and how the features directly related to making a song popular change.

# **INTRODUCTION**

There are so many potential reasons to make a song popular, maybe because the artist is super popular, maybe the song beats and liveliness and energy level is super high, which draw people’s interests. Or maybe the songs are in the genre that super popular in the period of time the information retrieved.

Main dataset we used is Spotify Tracks DB [1]. Other dataset we used to compare are SpotifyAudioFeaturesNov2018 [2], another one SpotifyAudioFeaturesApril2019 [3], top2017 [4], top2018 [5], top2016 [15], top2019 [16]

We are going to use Jupyter notebook[6], Pandas Dataframe[7], statsmodels.api[8], Scipy[9], seaborn[10], pyod.models[11], and matplotlib.pylab[12] to perform out data analysis.

* 1. **Motivations**

The question we want to answer is what features are directly related to making a song popular, is there a pattern? or does it keep changing? How have the trends been changing over time?

We think this would be useful for music producers and writers to see the trends between the most popular songs, this will help them have a better idea if the song they are producing or writing has a good chance in the top charts. Also, they will be able to see with our analysis what artists have the most amount of hits and which artists are most likely to produce hits according to our predictions. This analysis can also help radio stations and music broadcast apps like spotify itself, that’s because we will be providing information on what the consumer wants to hear more of, what features in a song are the most appealing to the masses and what features are not.

* 1. **Challenges**

For this specific dataset, analysis has been done on now single features affect popularity and visualization of the top popularity within each features

This dataset contains 30 millions songs with 17 features for the year 2018. There are a lot of noisy data will be involved, and at the same time, the time duration is restricted. We need to be able to take our the features that are highly correlated to popularity, and at the same time, we need to be able to analyze the outliers to see whether they are just noise or there are significant pattern related to certain outliers to make it popular.

* + 1. *Dataset Time-Duration Restriction*

This data set we have restricted to the time duration of a year. All the popularities and features are in year 2018. In order to make sure our analysis is not restricted to one year period, we will need to perform compare analysis to another data shares similar features but in a different time period.

Since all the songs are retrieved during 1 year time duration, our analysis might be biased if we only work with single dataset. Moreover, the dataset might not be representative enough without compare with other dataset.

* + 1. *Find Correlated Features Efficiently*

There are 18 columns in the dataset, which means 17 features except popularity. Doing single feature correlation with popularity is ok, but once we want to combine more than 2 features, there are too many possible combinations of different features we need to try out to see what works and what doesn’t. Even though we can select single features better correlated to popularity, it is still not efficient to find all the combinations and check the correlation.

* + 1. *Outlier Analysis*

Outliers might be noises or it might be significant samples that provide us potentially more knowledge of information about what makes a song popular. Since the dataset itself if big enough, it is hard to perform outlier analysis for the dataset as a whole. Need to find ways to break the whole dataset in parts (e.g. genre) and then perform outlier analysis to see what makes them “standout”.

Need to be selective when doing outlier analysis. If there are certain features do not necessarily impact popularity, the outlier analysis doesn’t actually affect the result.

* + 1. *Finding Top tracks of 2016 and 2019*

We could only find datasets of top tracks for 2017 and 2018 which had most of the same features as our original dataset. But we wanted a couple more datasets in order to clearly observe any trends or patterns in the music features throughout the years. We found an official website of Spotify that shows the top 200 songs of a given week from 2016 to 2019 and csv’s with those datasets. What we did was download the csv’s of two datasets, one from 2016 and another from 2019, which just contained the name of the song and the artist. Once downloaded, we filtered out the rows from our original dataset that contained the songs from our top songs of 2016 and 2019 and put these in new datasets. We were able to look up the top songs of 2016 and 2019 in our original dataset because this dataset was last updated 4 months ago, and contained most of the songs from the top hits of 2016 and 2019.

* + 1. *Other Challenges*

We want to make sure that the information we are extracting from our analysis is as representative as possible; that means making sure that other datasets haven’t done the same exact analysis, and find highly correlated features we can draw a conclusion from.

Another thing we need to take into consideration is that the music industry is a business that is constantly shifting and that means that what is popular one year may not be popular next year, so we want to make sure to test our model with another test set relatively in the similar time span first, and take another dataset into account or do compare analysis with other datasets

* 1. **Contributions**
     1. *Mengchen Gong*

Data Clean:

* Remove nominal columns with too many distinct values
* Nominal data - average, ranking;
* Numerical data - scatter plot

Data Visualization:

* Before and after Simplification
* Scatter plot, boxplots heat map

Feature correlation:

* Heat Map

Combined feature correlation:

* whole dataset and selected attributes and sub dataset

Outlier Analysis:

* Finding the trend of whether outliers are noises or useful evidence
* Scattered Plots

Data-simplification:

* simplified numerical data into groups,
* change nominal data to numerical data for future usage

Trend analysis:

* After simplification, dig more into features to see what character of the feature is better positively related to popularity
* Top genre analysis (before simplification and after)

Popularity Prediction (binary) Linear Regression:

* First step of prediction.
* After trained on 80% of the data, test on 20% test set to predict a song to be popular (1) or not popular(0).
* Used Logistic regression, Sigmoid function, Gradient Descent, and weights with bias.
* Used the trained weight to predict popularity class
* Multilinear Regression using statsmodels [8]
  + 1. *Juanfran Ferrer Peasley*

Original Dataset Analysis:

* Popularity mean per Genre: Found what the popularity average is in each music genre and made a visual representation of it
* Genre Correlation: Given that a song has high value on a certain attribute, the changes that the song is popular in each Genre.

New Datasets Analysis and Comparison [2], [3]:

* Correlation heatmap on new Datasets: Found two new spotify datasets with most of the same attributes and did a correlation heatmap
* Scatter Plots
* Box Plots

Popularity Trends 2016 - 2019:

* Finding and Filtering top 100 songs of 2016 and 2019: There were only lists of top hits of 2017 [4] and 2018 [5]. Found list of top hits of 2016 [15] and 2019 [16] and filtered those songs out from the original dataset and placed them in new dataset
* Correlation Heatmap of each year
* Top artists each year: By the amount of songs they have
* Unique songs each year: Filtered out the songs that any 2 years in a row had in common, in order to obtain better results
* Displots of the popularity by year of each feature
* Graph: Evolution of popularity of each feature

Popularity Predictions:

* Notable music trends over time: What % has each feature increased or decreased over time, and if its increasing or decreasing right now.
  + 1. *Carlos Prieto Fernandez*

Feature correlation Analysis

Implementation and analysis of different probabilistic models to determine whether the popularity can be predicted or not:

* Logistic Regression Model using scikit learn libraries [17] to fit.
* SGD model using scikit learn libraries[18] to fit, find the best features and model evaluation.

Model Evaluation:

* Recursive Feature elimination in order to get the most effective features.
* Confusion Matrices
* Roc Curves
* Classification reports which include: precision, recall, f1-score, support

# **RELATED WORK**

### Work has been done before on this dataset is showing in this link <https://www.kaggle.com/zaheenhamidani/ultimate-spotify-tracks-db> [1]. Analysis has been done for this dataset mainly focusing on popularity for different tracks, different artists, and popularity trend visualization [13]. There are similar datasets on Kaggle related to spotify music have similar approaches that analyze the popularity among single features, and data visualization in histograms [14]. What makes ours different is that we are going to not only see if there are positive/negative effects between different single feature and popularity, we are also going to analyze the correlation among features, and moving forward we will be able to use the learned knowledge from our analysis to estimate the probability that a song becomes popular in the near future.

* 1. **Most popular track**

The value of popularity is between 0, for least popular, and 100 for most popular. And the popularity score is based on the number of total number of playback for the tracks. By summing the popularity score for 3 different time period, they calculated the final popularity by taking the mean of the popularity sum.

* 1. **Top Artists by Popularity**

On this previous work it mainly shows the top 20 artists by number of tracks in the top 100, so it ranks them in order to see which artists have more top songs. It also makes a ranking based on the total amount of an artists ́songs popularities all added up.

* 1. **Popularity Visualization**

On this previous work what they did was plotting the values for popularity for the top ten most popular songs in the summer against the time span of the summer months to evaluate the evolution of the popularity of the songs

# **METHODOLOGY**

* 1. **Feature Popularity Correlations**

### Having 18 columns (features) in our dataset, we will try to find out whether each feature has positive or negative impacts for the popularity. We will simply isolate out each column and do data analysis with the popularity column.

### Then we will do scatter plot for each feature to see if there is a correlation that exists between the feature and the popularity. If so, we will see if that is a positive correlation or negative correlation.

* 1. **Feature Correlations**

After the feature-popularity correlations have been done, we will step forward and trying to see the correlation among features. For 2 features we are trying to find correlations, first, we will do z\_score normalization for both features, then we will plot them in 2 side by side plot. Normalization will help us see 2 different ranged feature data a lot clear.

Additionally, we will calculate the correlation coefficient between 2 chosen features. Depending on the result, we will know if 2 features have strong agreement with each other (correlation coefficient close to 1), strong disagreement (correlation coefficient close to -1), if there is no strong agreement or disagreement among those 2 features (correlation coefficient close to 0).

* 1. **Multi-feature Popularity Correlations**

### Knowing the features we have for positive/negative correlations with popularity, and agreement/disagreement among features, we can do further analysis on grouped features.

* + 1. *Positive/Negative Correlation Features.*

### Divided all features into 3 groups, first the ones with positive/negative correlations, we will perform analysis on how all positive/negative features will affect popularity. There are a couple approaches we can try here, the first is normaliza all features data and then take the mean of all positive/negative features for each track. This approach only works for numerical data. Second, for binary data type, since there is only 1 binary feature (mode), we don’t need to do further analysis on it. As for nominal features, we will do group by and do a simple analysis to see which category in that features has the strongest positive/negative correlations with popularity.

* + 1. *High Correlation Features Group.*

### Based on what are the features in 3.1.2 has strong agreement/disagreement relation with each other, we will see how those features together correlate to the popularity.

* + 1. *Multiple Regression Model*

What we are trying to do here is build a multiple Regression Model which looks like y =b ₀+b ₁x ₁+b₂x₂ where the value to estimate “y” is the popularity and x1 and x2 would be the values for two features from our data set. For the constant values b ₀, b₁ and b₂ we calculate them using a library that gets the slope and intercept of the line best fit for your model. Once we have our model we have tested it on our dataset and used the Coefficient of Determination so we could evaluate the results of the different models that we generated using different combinations of features.

* 1. **New Datasets**

3.4.1 Obtaining new datasets

In order to prove the Spotify dataset we have to be reliable we wanted to do some of the same analysis we did in our original dataset on other datasets, which must have at least most of the same features. We obtained two new datasets that contained almost all of the same features. They were just missing music genre, but this wasn’t necessary to prove if our datasets are reliable

3.4.2 Correlation Heatmap

We do the correlation heatmap on both new datasets and compare it to the original dataset. If we get similar results, or slightly shifted since these datasets are from different years, then we can prove our original dataset is reliable. If we happen to get very different results then we should look for other datasets dataset.

* 1. **Genre Compare Analysis**

We wanted to see what was the correlation between the music genres, the song features (danceability, energy, acousticness, liveness and instrumentalness) and the song popularity. To do this we calculated all the rows that contained that certain genre and had a high value of the feature (higher than its average) and we divided this by the amount of rows that contained that genre, a high value of the feature and were also considered popular (popularity was higher than average). By doing this we are able to estimate what were the chances that a song is popular given its genre and a dominant feature of the song.

* 1. **Train-Test All Data**

After done above analysis, we have known about knowledge of single features. We will split our dataset into 80/20, 80% training data and 20% testing data. All the training and correlation finding are performed in the 80% training set, and then we will use the knowledge we mined from training data to make predictions on the 20% test data to see whether it is close to the original given classes. If the conclusions we found from the 80% are not close to the 20% for testing we will go back to analysing our 80% to see what we did wrong. We also want to make sure to report anomalies in the analysis of our dataset so it doesn’t mess with our conclusions.

To confirm what features affects the popularity the most when combine together, we first use all features initial an all-zero weight vector for all features. The length of the vector will be the number of features with an extra bias term, which also set to 0. Set the learning rate to 0.001.

* + 1. **Binary Logistic Regression**

We will first perform binary logistic regression for our data. Begin with splitting popularity in 2 case, popular (50-100), unpopular (1-50). If a song it’s popular, the score will be 1, on the other hand if a song is categorized unpopular, the score will be 0. Train the weight vector based on 80% training data and update weight vector along the way. At the end, the goal is to be able to use the trained weight to predict songs in binary category.

* + - 1. *Sigmoid for Prediction*

Using sigmoid function to get the probability assigned to popular/unpopular case between 0 to 1. If the predicted probability is bigger than 0.5 then the predicted class will be 1, which is popular. Otherwise, the predicted class will be 0 as unpopular

* + - 1. *Cross-entropy Loss as Loss Function*

Use cross-entropy loss to see how far the prediction is away from the true class, then based on the loss, we will update the weight vectors. For weight vectores, if a weight is positive, it means it is positively correlated to popularity. If a weight is negative, means its negatively correlated to popularity. If a weight is close to 0, it means it doesn’t affect popularity that much.

* + - 1. *Gradient Descent*

Use gradient Descent to update the weight based on the predicted class () and the true class (y). Here, true class y is either a 1 or a 0. 1 stands for popular (popularity between 50-100), and 0 for unpopular (popularity 0-50). We update weight vectors for each sample in the training data.

* + 1. **Softmax**

There is no hard line between popular and unpopular, a multinomial regression model fits better in this case. So rather than have popularity score 1-100, popular vs unpopular for binary logistic regression. We will divide popularity into 5 categories: A, B, C, D, E, which represent for popular, popular -, medium, unpopular +, and unpopular. And apply Softmax on it so predict which class a song will fit in.

From the 80% training data, we randomly choose 1 track out, for each randomly drawn track, we will use softmax multi-class classification and gradient vector to update the new weight vector. Similar to logistic regression, we will still use cross-entropy loss as our loss function and combined with gradient descent to update the weight vector.

* + 1. **Testing**
       1. *Testing Use Trained Weight.*

## We will perform on the 20% test data that hasn’t been touched. On the original dataset, the popularity are specific numbers rank from 0 to 100, like mentioned above, we will do 2 tests. One for the weight trained by logistic regression, and another one test on weight trained by Multi-class model, trained by softmax. For linear logistic regression, we will see whether the model is able to predict popular vs. unpopular. For multi-class model, we divide popularity into 5 different categories, so 0-20: E, 21-40: D, 41-60: C, 61-80: B, 81-100: A. We will test on the testing set to see if we can correctly put tracks into the right category.

* + - 1. *Testing Use Different Dataset.*

## We will compare our results with the top popular songs of the dataset to see if our result is close or somewhat related (based on the genre, danceability, and so on) We will also look at different popular charts from different radio stations and compare it with our results and see if we were on the right track. We will also be able to determine how much of a chance a song has to be in the popular charts given our assumptions from the dataset.

* 1. **Train-Test Selected Data**

## Train on the whole data is cool, but we want to take a good look at how features affect popularity for certain attributes. For example, we want to see what are the features that make a top artists popular. Similar, for different genre music, what are the features make top genre popular, are they in common or each genre has different features place important roles.

## All work like training and testing are use the same method mentioned in 3.4, the only difference will be train and test in a subset of the whole dataset.

* 1. **Outlier analysis**

## Isolate out the outliers for selected data based on different features, based on the trend and the visualization of the outliers (both at the higher end - high popularity, and lower end - low popularity), we want to see if some of the outliers are noises or significant evidence of why songs are popular/unpopular.

* + 1. *Outlier for Certain Features*

For selected features (subset of dataset), we are going to analysis the features correlations for the outliers. Outliers here correspond to the songs’ popularity is higher than 1.5 IQR range or lower than 1.5 IQR range for boxplot. Since what we care about what makes a song popular, we will first focus on the outlier songs whose popularity is higher than normal (higher than 1.4 IQR from Q3). Take subset of movie genre for example, here the outliers will be the move genre songs whose popularity is 1.5 IQR higher than the 3rd quartile of the value.

After getting the outliers for certain attributes, we will analyze then by checking the correlation between single feature and popularity to see if we can find a pattern for all of the outliers or part of the outliers. For the ones out of normal and couldn’t find a pattern, those samples are likely to be noisy data points. On the other hand if there is a pattern showing, it might provide us some useful information or insights of why those songs stand out in the attributes.

* + 1. *Comparable Outliers Analysis*

Retrieving results from 3.6.1, we will have a ground knowledge of what makes songs in different attributes have outstanding popularity. Among different attributes’ subset, perform a compare analysis will be super useful that see are same features make popularity stand out for different attributes.

If there are common reasons for those out-perform songs, we can make a conclusion about what makes a song popular. If there are not common features make song more popular than normal, then it is a case-to-case situation that certain features will make the songs in certain attributes specifically more popular.

* 1. **Simplify Data with Analysis**

For numerical columns, except popularity has scored 0-100, rest of the numerical columns has 0-1 values, there are too many different values for each feature that when doing analysis, which bring hardness to analysis.

* + 1. *Simplify Numerical data*

Because of that dataset simplification has been applied. Leave Popularity untouched (we will group popularity at the later step), for all other features, all columns has been simplified from 0-1 to 1,2,3,4,5 five groups, which are 0-0.2, 0.2-0.4, 0.4-0.6, 0.6-0.8. 0.8-1.0 respectively. Regrouping data will be able to help us to see the better more in detail trend in each of the subgroup.

* + 1. *Simplify Nominal Data*

In order to be able to show the correlation of nominal data with popularity, change the nominal data into numerical data will be important. For this part of the analysis, we deleted the column artists name, track name and track id because those column has too many unique values. But for key, mode, time\_signature, it contains a small number of unique values and it will be helpful to convert into numeric.

* + 1. *Analysis for Simplified Dataset*

After simplification, we are able to do data visualization on the features to popularity in a group base. Boxplots here will be super useful to help us to see how different groups of data in a feature will affect the popularity, or which group of a feature plays an important role to make a song has higher popularity.

* 1. **Top 5 Popular Genre Analysis**

For top 5 popular genres (highest mean popularity), which are Pop, Rap, Rock, Hip-Hop, Dance. Want to do analysis to see what are their feature correlations to the popularity. Then take an even closer look at the songs in each genre (popularity>80) to see how the correlation change and give us more evidence of what make a song popular, there might be things in common and also things different based on the genre’s difference.

* + 1. *Top Genre Popularity Correlations*

The top 5 genres that has highest mean popularity will provide more information about what makes them popular. Sub-heatmaps are generated for each of the genre.

* + 1. *Top Genre Popularity>90 Analysis*

The heat maps produced in 3.9.1 showing a broad overview of what is positively correlated what is negatively correlated or what is not affecting the popularity. This step is performed after simplify (regroup) the data, multi-linear regression is performed after simplification and for each of the top genre, coefficients will be evidence of what play important role to popularity high,

* 1. **Evolution of popularity 2016-2019**
     1. *Why do this?*

After we finished the analysis on our Dataset and the new datasets, we wanted to see how music tastes have evolved through time. Unfortunately, the latest popularity information that can be retrieved from Spotify is from 2016, but fortunately this was enough to extract good information and come to some conclusions. With this analysis we are able to find how trends shift throughout the years and what people gravitate to over the years. This is a key element in order to predict what the future trends are most likely to be like

* + 1. *Obtaining the datasets and cleansing Data*

We were only able to find two datasets of the top 100 songs of that year (2017,2018) that had similar attributes to the ones that were in our original dataset. Some information can be extracted from these two years but we felt like we needed data from more years in order to properly analyze it. We were able to find an official Spotify website that had csv’s containing the top hit songs from each year from 2016-2019. So we downloaded the csv of 2016 and 2019 since we already had the top songs of 2017 and 2018. Once downloaded we noticed a problem, that is that we had the top songs of 2016 and 2017 but we didn’t have the features of each song. To solve this, what we did was search the top songs of 2016 and 2019 in our original dataset (which is from July of 2019 so this can be done), filtered out the rows that contained these songs with their respective artist and put them in a new dataset, one for top songs of 2016 and another for top songs of 2019. Now these two new datasets will have the same attributes as our original dataset and most of the same attributes (at least all of the ones we care about) as the datasets of the top songs of 2017 and 2018. The only thing we had to change after this was the title of two columns in the 2016 and 2019 datasets in order for them to match the titles in 2017 and 2019. We changed ‘track\_name’ to ‘name’ and ‘artist\_name’ to ‘artists’. Now we could begin the analysis.

* + 1. *Filtering the Unique Hits of each year*

We noticed pretty early on that some of the same songs could be found in the top hits of two consecutive years. We wanted to filter these out in order to get better data of the popularity of the songs that year. In order to do this we found the common hits of 2016 and 2017 and filtered these songs out of our current datasets for those years, we did the same thing for 2017-2018 and 2018-2019.

* + 1. *Correlation Heatmaps*

The first analysis we did was to perform a correlation heatmap, this will tell us what features are directly related and which aren’t in making songs popular.. It's also good to do this in order to see if there is anything that really stands out about the correlations.

* + 1. *Artists with most hits each year*

We wanted to see what were the artists that had the most hit songs each year, so we filtered each dataset by the number of songs per artist. This can also help us see what artists keep making multiple unique hits in consecutive years and which ones fade out. We displayed the artist of each dataset that had more than one hit.

* + 1. *Visual representation of the popularity of each feature throughout the years*

We wanted a visual representation of the popularity of each feature every year in order to get a better understanding of the change in trends. We did a displot for every feature and plotted the popularity of that feature of each year.

* + 1. *Graph of evolution in popularity of each feature*

We wanted to see how the trends have been changing each year, how each feature has changed in popularity over the years. This is key to see patterns in trends and be able to predict future trends.In order to do this we calculated the mean value of each feature for each year and made a graph for every feature where we plotted this.

* + 1. *Percent of increase or decrease of each feature from 2016-2019*

We want to know how much percent has that feature changed from 2016 to now, so to do this we divided the mean value of each feature in 2019 over its value in 2016 to see how much its increased or decreased.

* + 1. *Future predictions for each feature*

Now with the visual representation of each feature over the years and the percent of change we are able to see if it's currently increasing or decreasing and how much its changed since 2016. This can give us a good estimate on whether the feature might become more popular or not.

# **EVALUATIONS**

In order to evaluate our work we will need to set some standards and use some tools to grade our work and see if we get the desired results.

It’s important to verify our results with other datasets or even with music charts to make sure that our results are correct. We will also be able to evaluate our own data by performing multiple tests with the 20% of the data. We will allow a small margin of error given the fact that some anomalies in what songs are popular can happen. If our conclusions are not close to what other datasets and music charts say then we will have to go back and review our analysis for the dataset. We will be careful in every step we take during the process of analysing our data, continuous evaluations of our dataset will be needed to make sure we don’t formulate wrong conclusions.

* 1. **Accuracy**

For evaluating our model what we will do is use the model evaluation models learned in class. With a confusion matrix we will evaluate the accuracy of our model and in order to do this we will also have to calculate the specificity, sensitivity and precision. By calculating the accuracy we will get the degree to which the results of our model outputs conform to the correct value or label. We could also contemplate using ROC tables to show the accuracy of our models.

* + 1. *Binary Logistic Regression*

For binary logistic regression, we are only predict the popularity in 2 groups 1 means popular, 0 means not popular. It turned out that it is not super effective for the data set that we got about 67% accuracy of the prediction. Since there is not a clean separation for popularity <50 and popularity>50, more model need to built that doing multilinear regression.

* 1. **Errors**

We will use cross-validation for training our model and in order to choose the best model we could get,we will look at the error measures more specifically the t-test. By looking at the error we will be able to determine if the output we get has a high chance of being correct or not. Adding this to our evaluation will help us choose the best model which will be a combination of accuracy and error minimization.

* + 1. *Binary Logistic Regression*

Repeatingly redo the 80/20 split for the training data and testing data, the average accuracy for binary class predicting is about 65-70%. Like mentioned above, because popularity pretty much evenly distributed in the range

* + 1. Top songs from 2016 - 2019 specific dates

The Top songs from each year were extracted around the same time in the year but not the exact same date each year, this might slightly affect the accuracy of the data we end up obtaining.

# **DISCUSSION**

* 1. **Feature-Popularity Correlation Whole Dataset**

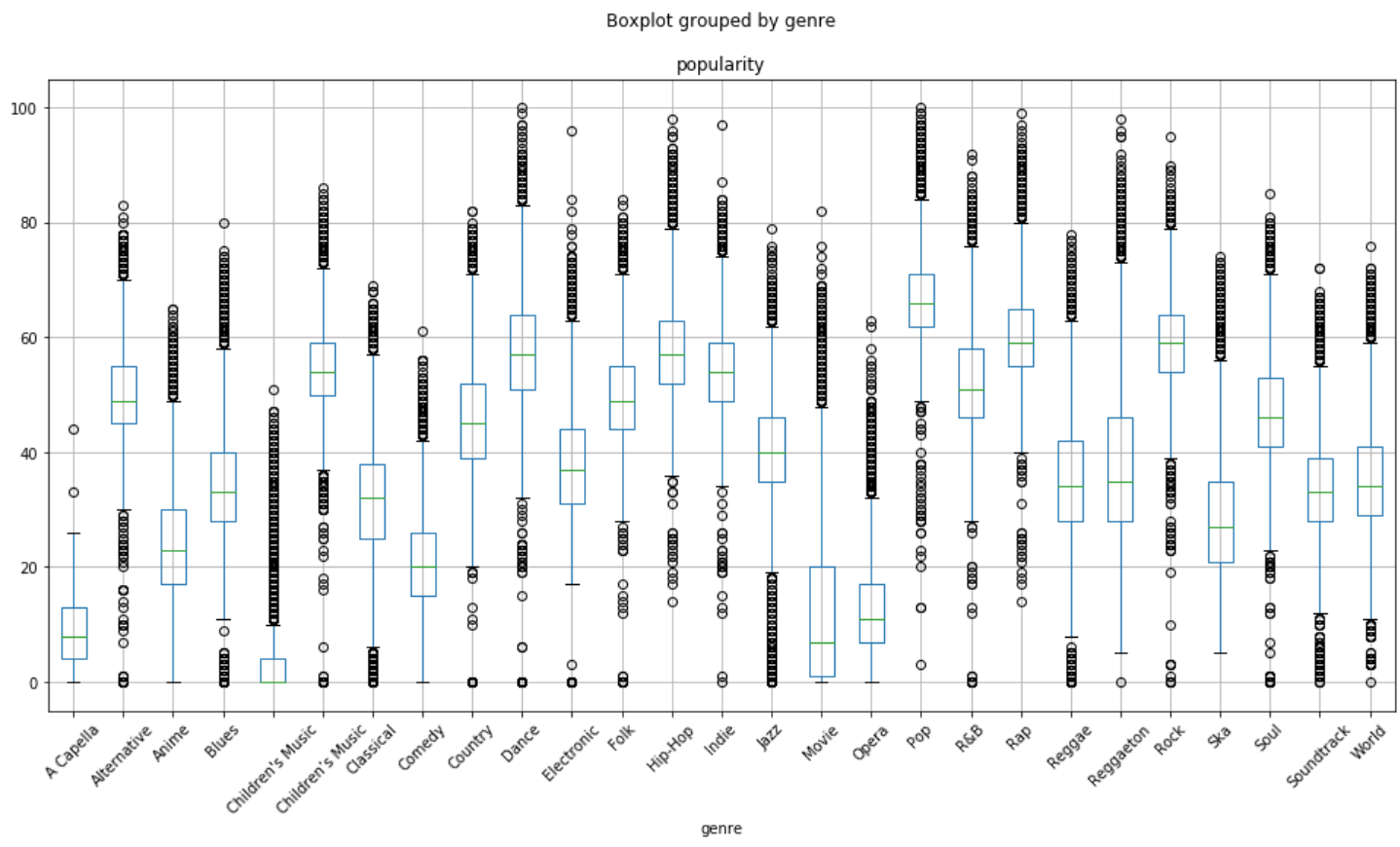
Solely looking at the correlation between 1 feature of correlation, Here are the visualization of how features correlate to popularity. For nominal data, boxplot is showing, and for numerical data, scatter plot is displayed.

As the graph shown below, we can see that by looking at the graph itself, it’s hard to see the correlations between single feature and popularity. What we can drawn conclusion based on this step is that features: key, duration\_ms, mode doesn’t closely correlated to the popularity. Hard to find trend solely in scattered plot for energy, valence, tempo Based on the boxplot, it is about the same for 12 keys in Key and major/minor in Mode. Except that, more analysis needs to be done to be able to see what feature(s) affect popularity after simplification.

For the features that contain characters or showing the trend of the correlations before simplify the data, scatter plot is shown below. For non-numerical data that have smaller number of distinct groups,, boxplot is shown below/

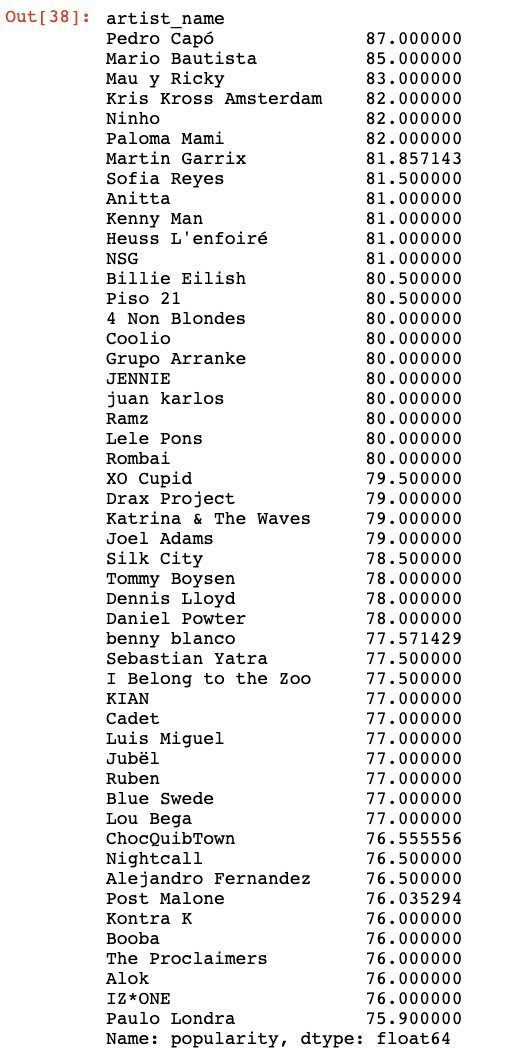
* + 1. *Genre*

We conducted means of popularity of each genre to see what genres were the most popular, the popularity range is from 0 to 100. We found out that the three most popular genres with their respective scores were: Pop = 66.59, Rap = 60.53, Rock = 59.62.

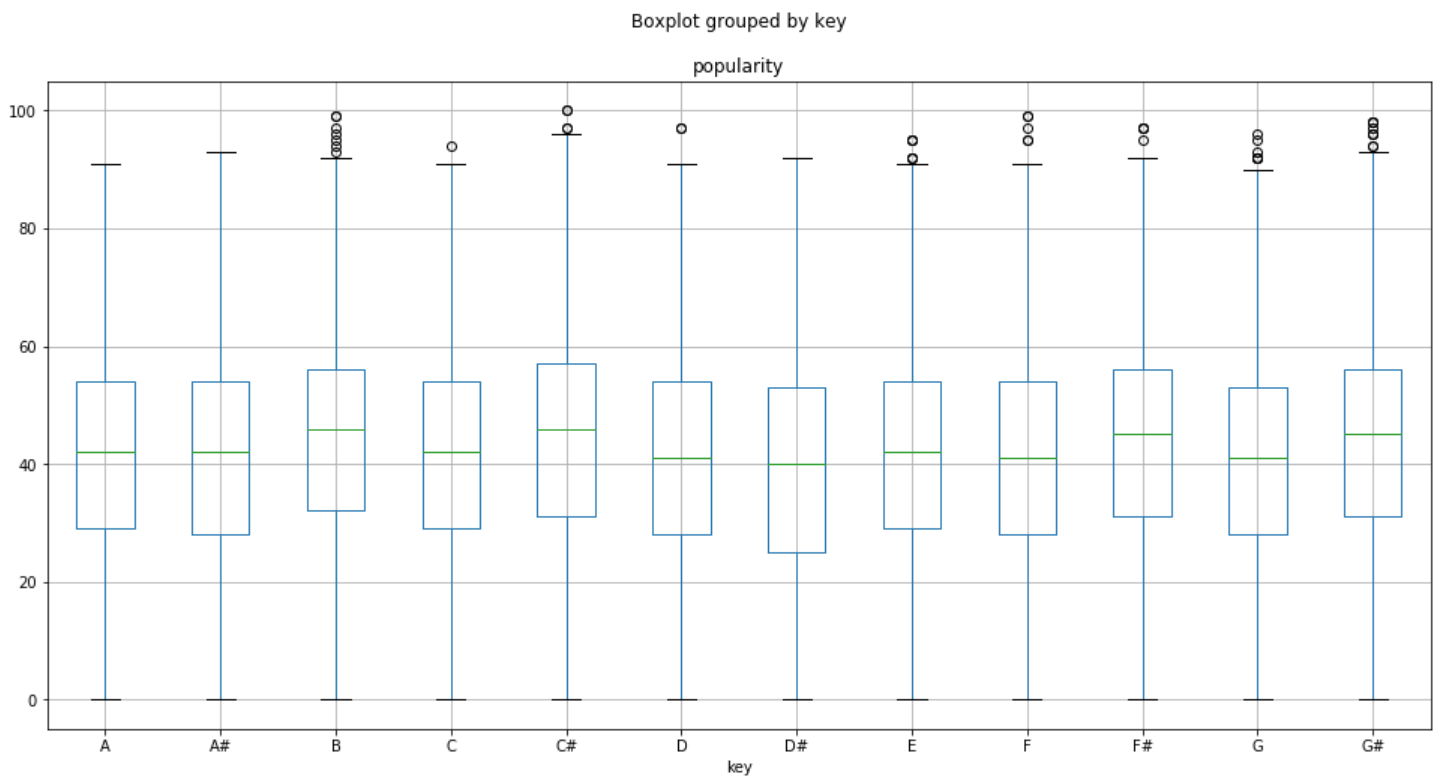


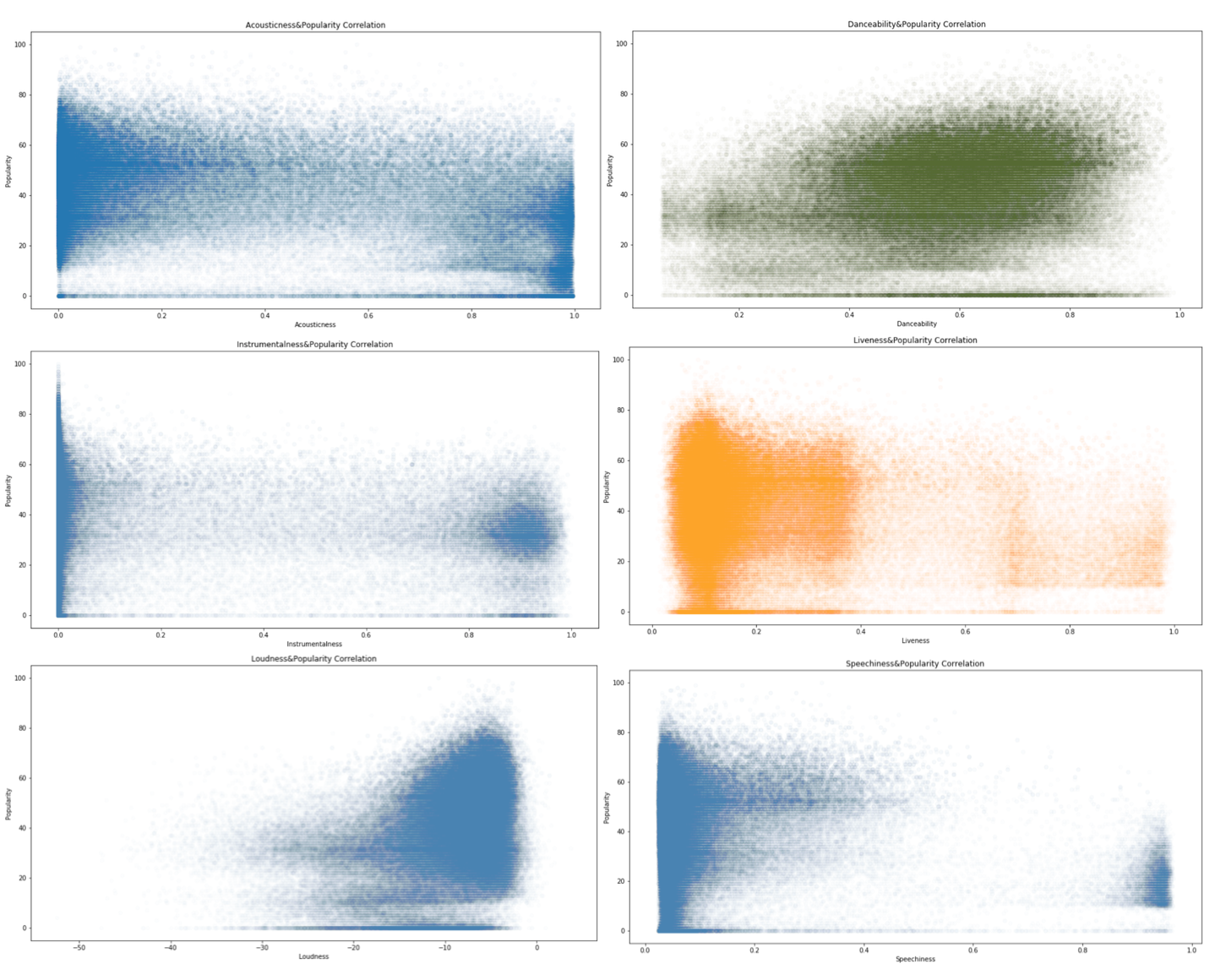
* + 1. *Artist\_name*

We calculated the artists with the most popular songs on average and this was the result:



* + 1. *Key*

There are only slightly difference among different keys for popularity, which means that key doesn’t affect popularity.

**

* + 1. *Track\_name & Track ID*

Those 2 attributes are too unique that we exclude from the analysis of feature-popularity correlation for this step.

* + 1. *Acousticness (left up)*

We can see that there is no huge correlations between acousticness and popularity but we can appreciate some really small negative correlation.

* + 1. *Danceability (right up)*

We can vaguely see the positive correlations between danceability and popularity.

* + 1. *Instrumentalness (left mid)*

We can say straight out of the bat that there is not really a correlation between these popularity and Instrumentalness.

* + 1. *Liveness (right mid)*

Only based on the graph, it is hard to tell how the trend is like.

* + 1. *Loudness (left low)*

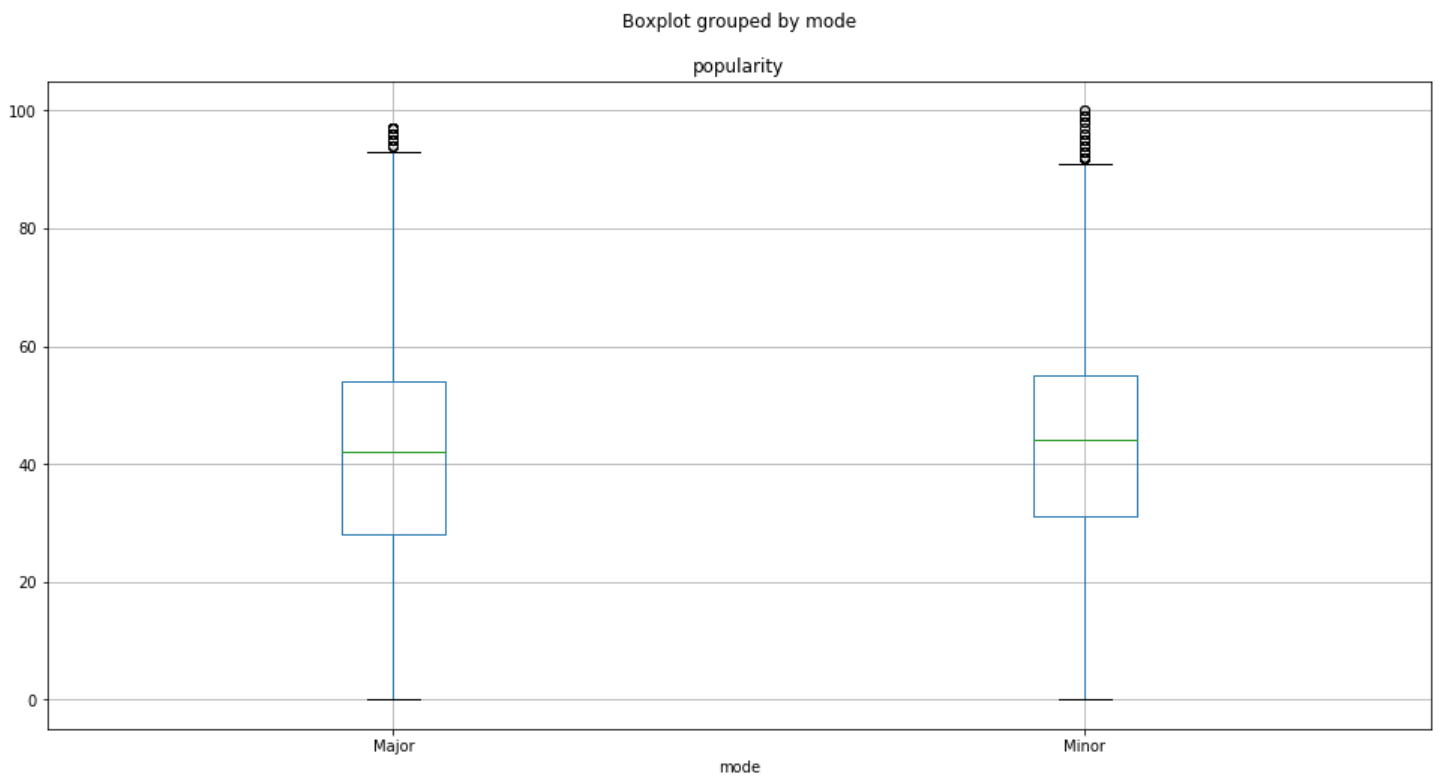
If a song is popular, the loudness level is high. However, on the other hand, we cannot conclude that if a song has high loudness, it is popular.

* + 1. *Speechiness (right low)*

Based on the graph, the more popular songs are, the lower level of speechiness appear. At the same time, less popular songs has higher level speechiness.

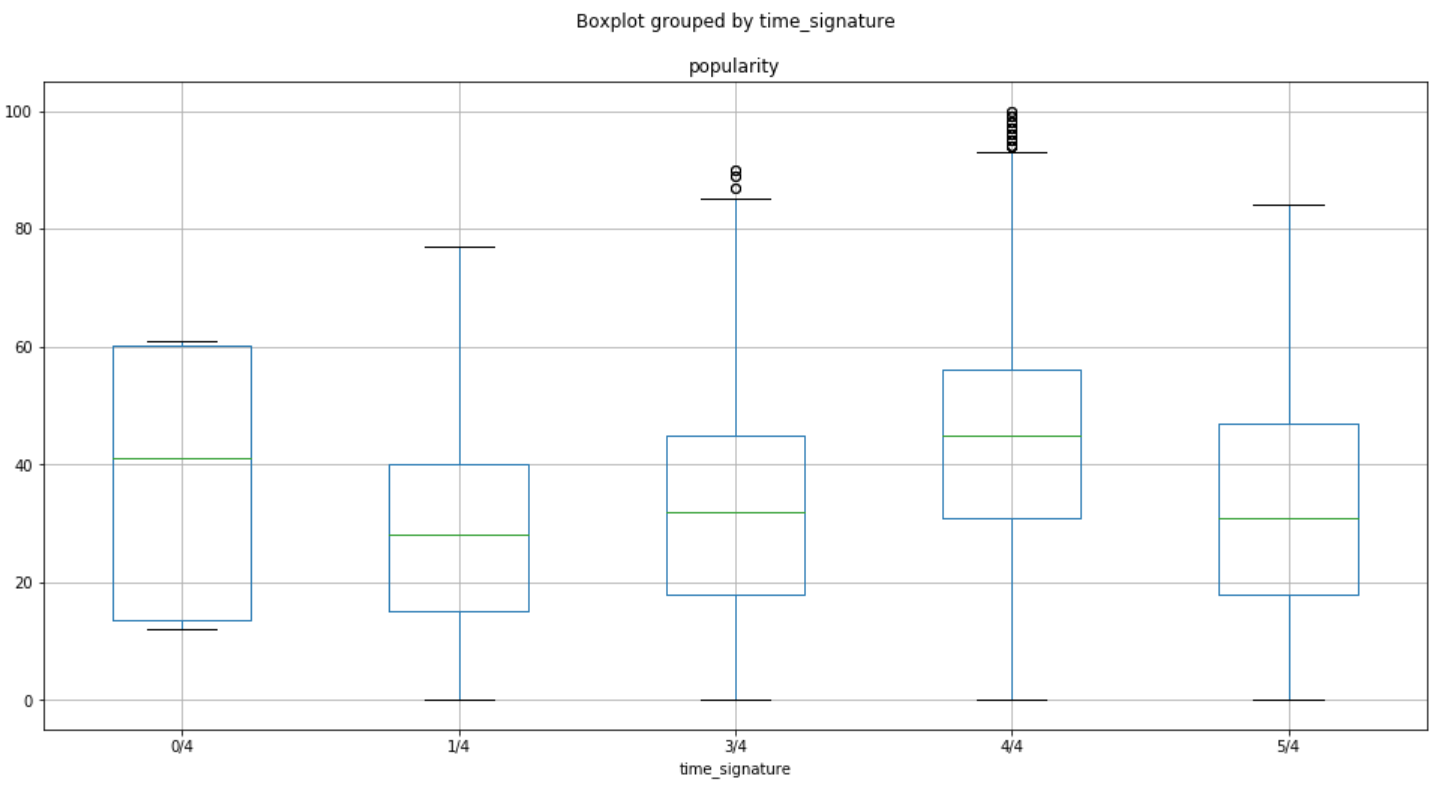
* + 1. *Mode*

Simply from the boxplot, both major song and minor song have the similar mean and similar distribution. There are more outliers in the minor songs, especially the outlier that has high popularity.

**

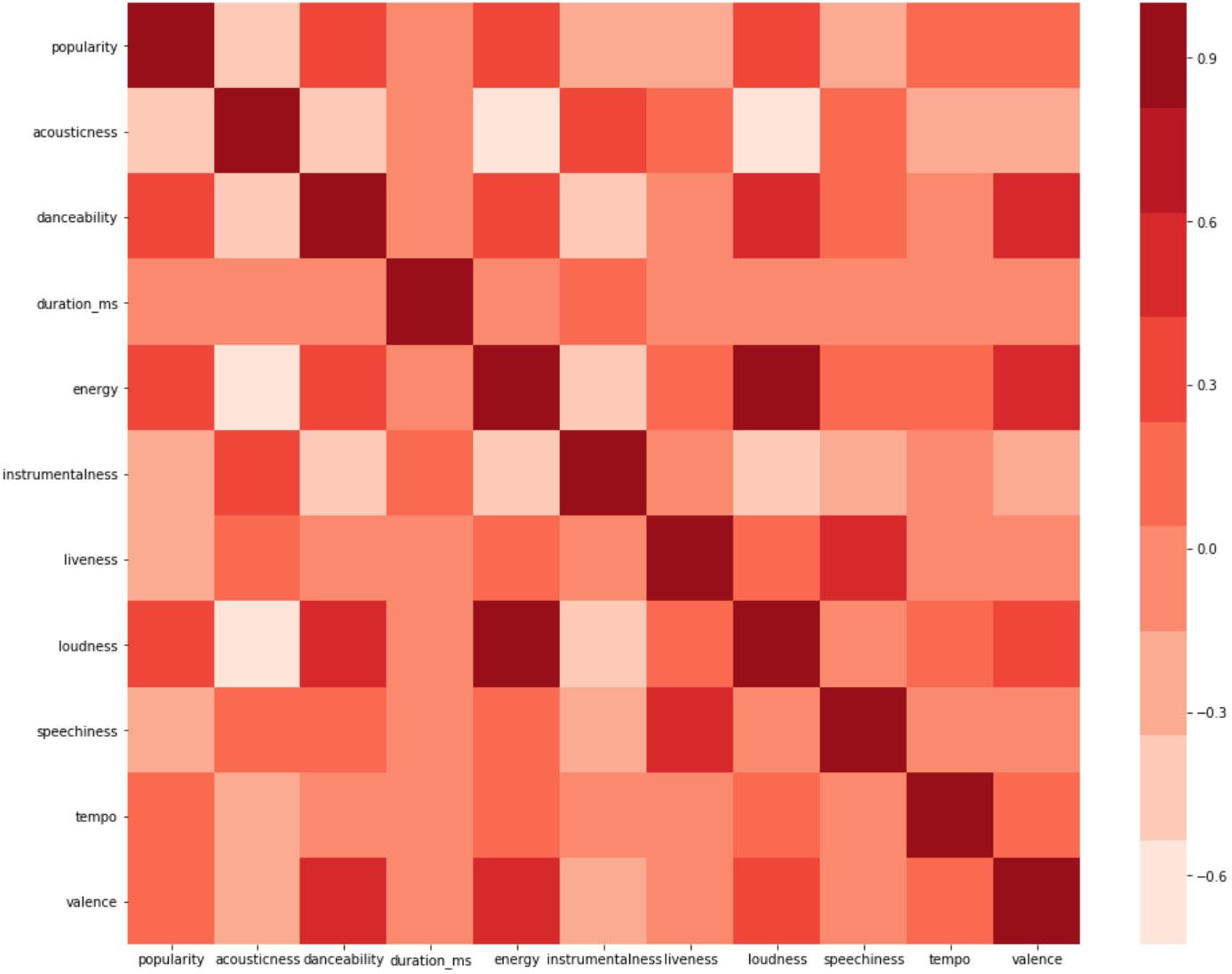
* + 1. *Time-Signature*

Shown from the graph, we can see that 4/4 has the highest mean popularity among all 5 time-signatures. More analysis needs to be done for the sub-group 4/4 time-signature potentially related to liveness and energy.

****

* 1. **Feature Correlation Whole Dataset**

This heatmap provide us a good overview of how features correlated with each other. The upper right half is the same as left lower half of the square.



Treat popularity as feature, we can tell from the map, danceability, energy, loudness, tempos and valence has positive correlations to popularity. Duration\_ms doesn’t affect popularity. Acousticness, instrumentalism, liveness and specciness as negative correlations to popularity. None of those correlations is super strong.

Among other features, loudness-energy has high positive correlation; energy-acousticness and loudness-acousticness has high negative correlation.

* 1. **Outlier Analysis**

Like mentioned in 3.7.2, trying to see the outliers in the dataset, which are the ones above Q3+1.5\*IQR or Q1-1.5\*IQR. Plotted out the low outliers using blue and plotted the high outliers using red and trying to see if it shows any trend for significant evidence.

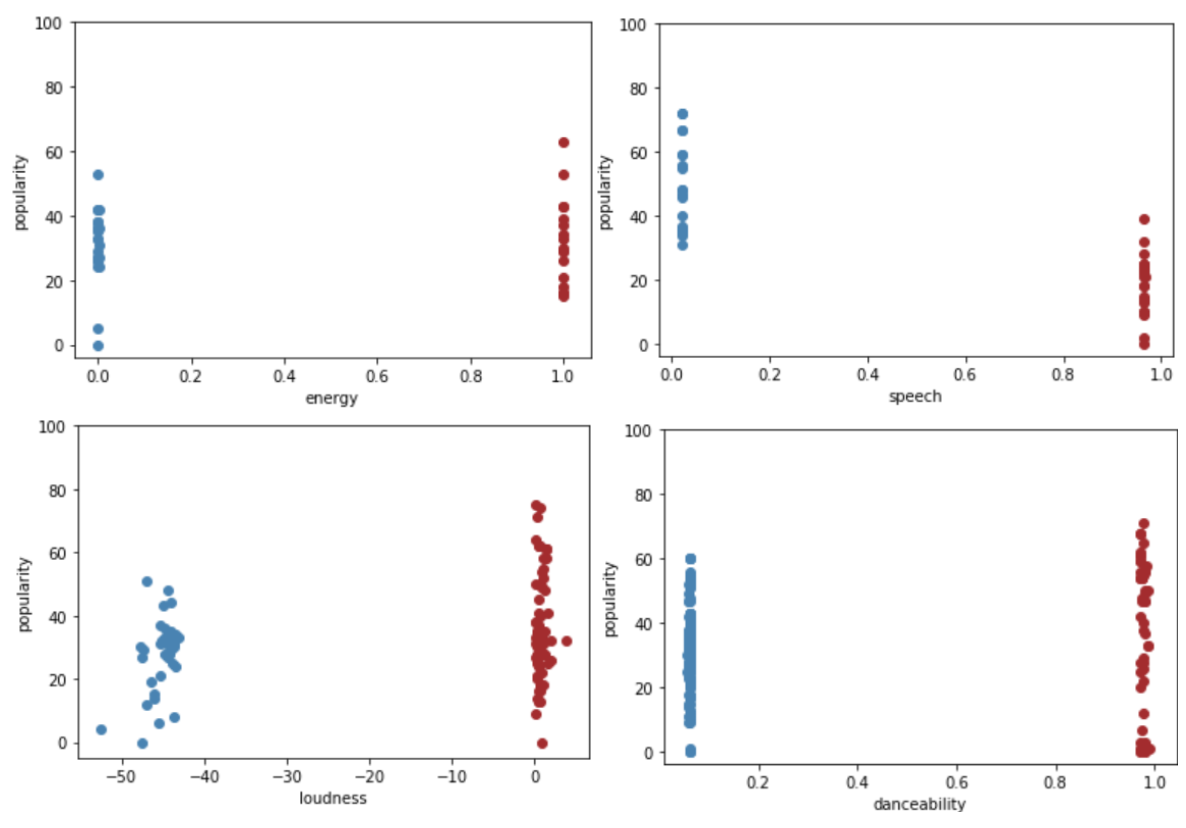
Based on the analysis, only some features’ outlier showing the evidence of why those extreme cases are happening, which are energy (left up), speechiness (right up), loudness (left lower), and danceability (right lower).

* + 1. *Energy (left up)*

### Outliers in feature energy has the trend that high energy has a better chance to have a higher popularity. Even though both high and low as condensed ones with the popularity around 20-40, but the popularity mean for high outliers are higher

* + 1. *Speechiness (right up)*

Speechiness outliers showing us an obvious trend that low speechiness ones have higher popularity. Majority of the outliers for speechiness are out noise but significant data points that provides evidence of what kind of speechiness within the song will make it more popular



* + 1. *Loudness (left low)*

Same as Energy, the higher the value, more likely for a higher popularity. For the low outliers, there is no single song with the popularity above 60.

* + 1. *Danceability (right low)*

Outliers for danceability showing less information, which is telling us that among those outliers, there are a significant number of outliers contained. The only things we can see from is similar to energy and loudness, high score of danceability does make high popularity songs more possibile.

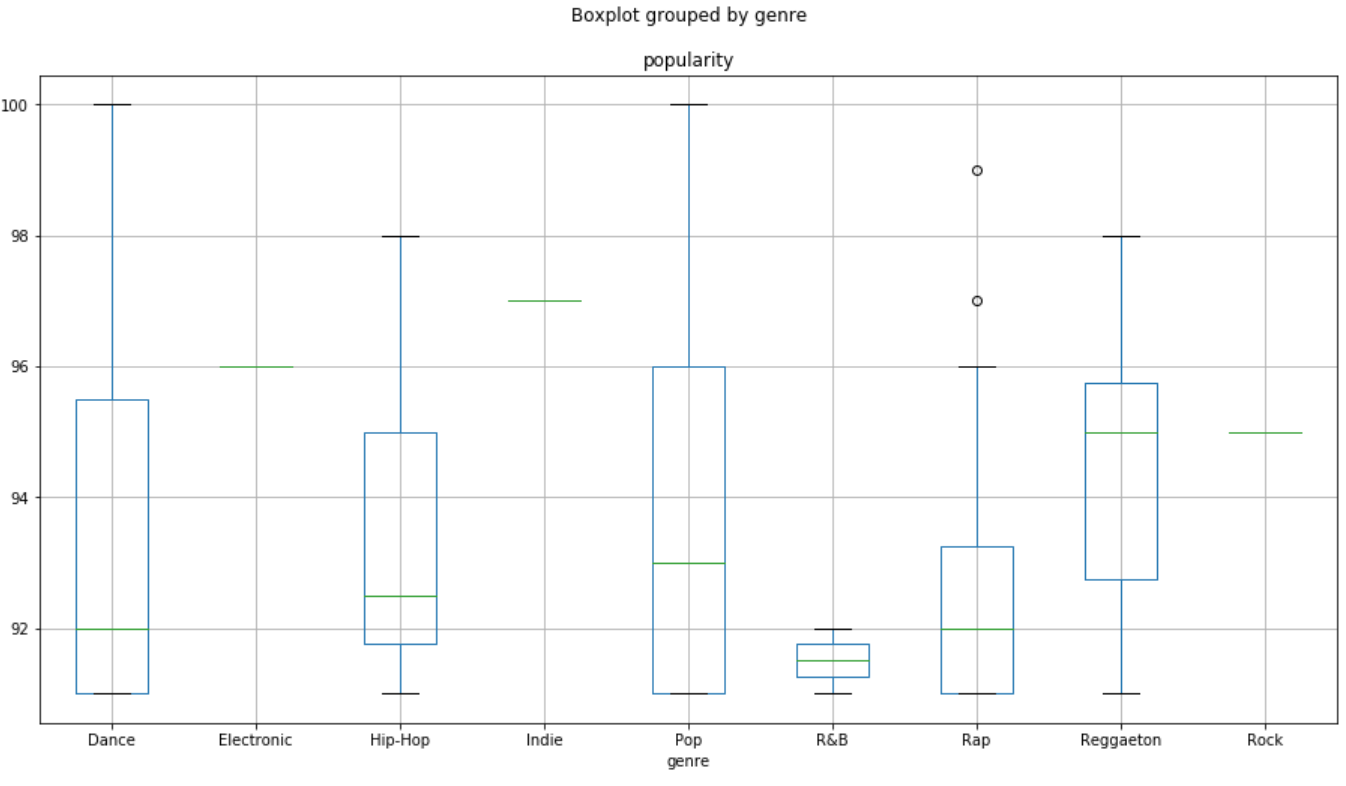
* + 1. *Others*

### The outliers for other features doesn’t quite show an obvious trend or there is not much difference between the lower outliers and higher outliers. More analyze will be performed after simplify the data.

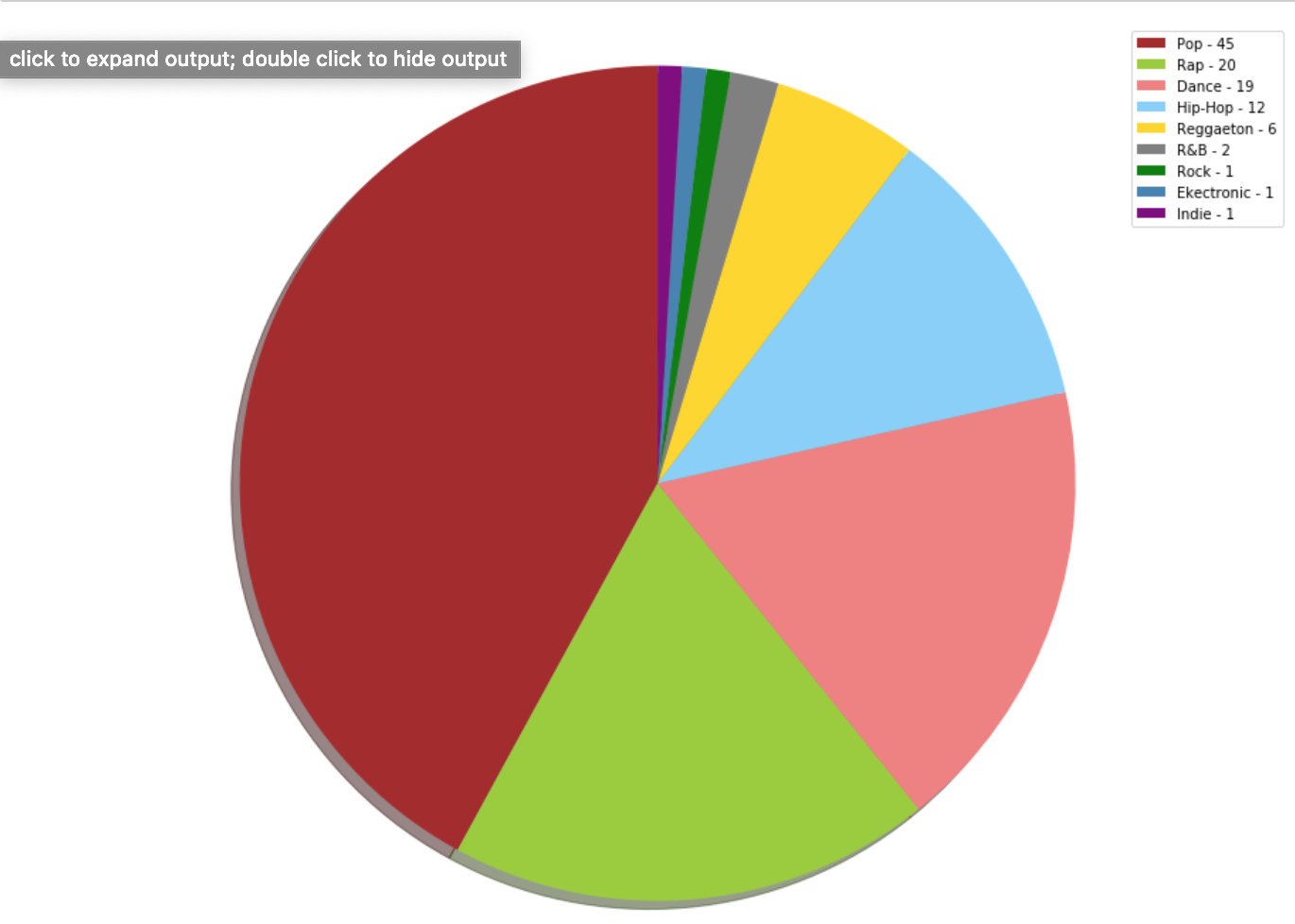
* 1. **Top Song analysis (popularity > 90)**

Isolated songs whose popularity is bigger than 90 out, then see how the distribution looks like.

Here are a boxplot and pie chart for songs popularity>90.



### Total 107 songs has popularity >90, among those, 45 are Pops, 20 are Rap, 19 are dance, 12 are Hip-Hop. Reggeaton, R&B, Rock, Electronic, and Indie has 6,2,1,1,1 respectively.



* 1. **Top 5 Genre analysis**

### The top 5 popular genre for the dataset judged based mean popularity are Pop, Rap, Rock, Hip-Hop and Dance.

* + 1. *Before Simplification*

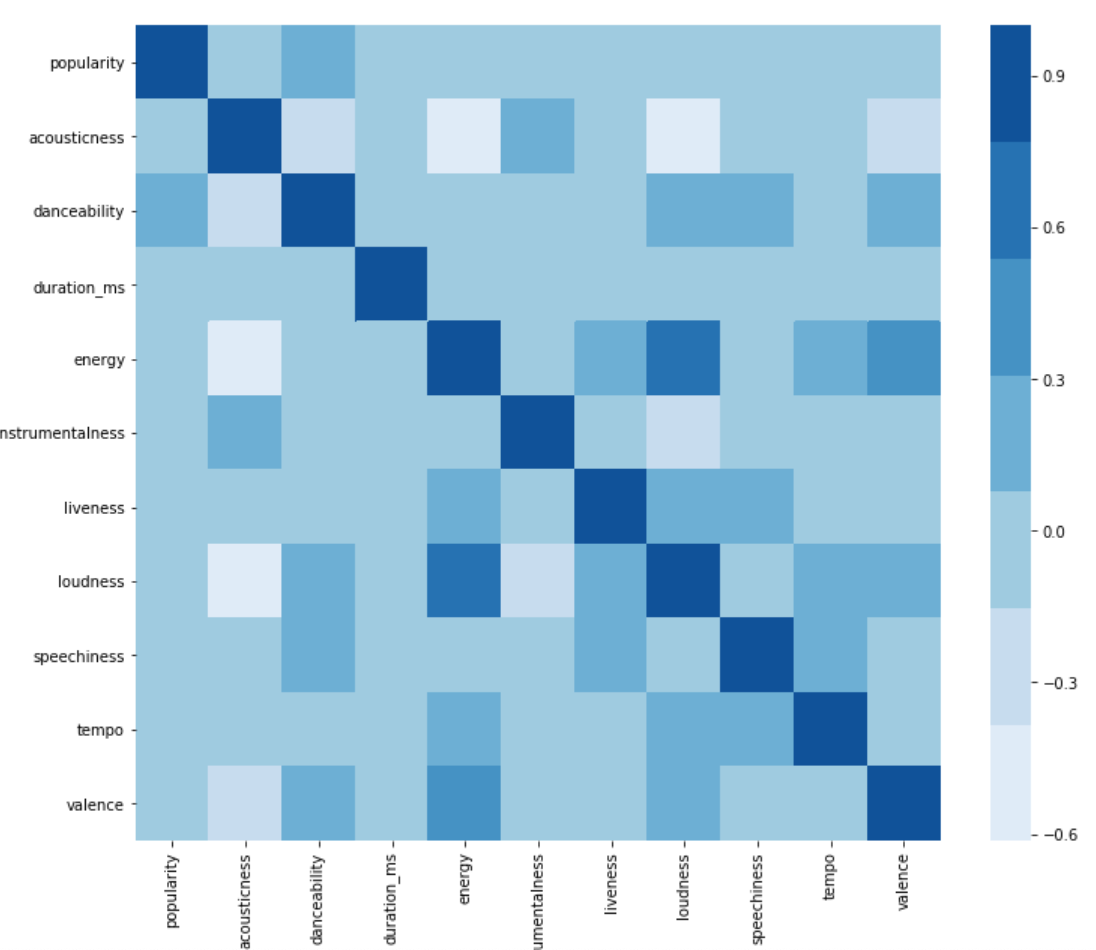
### Main method used here is heatmap. Which will give us a clear view of what features are correlate closely to popularity and at the same time what are feature-feature correlations.

### Heat map is a good way to show the trend, but the limitation here is that if there are small positive coefficient for some of the features, it is not going to tell only from the heatmap, because the color has been grouped from 0-0.3, or 0-0.1, the color doesn’t exactly represent the correlations.

### More work needs to be done to be able to see the coefficients for each of the feature, and it will be done in the later section after the simplification of the original dataset

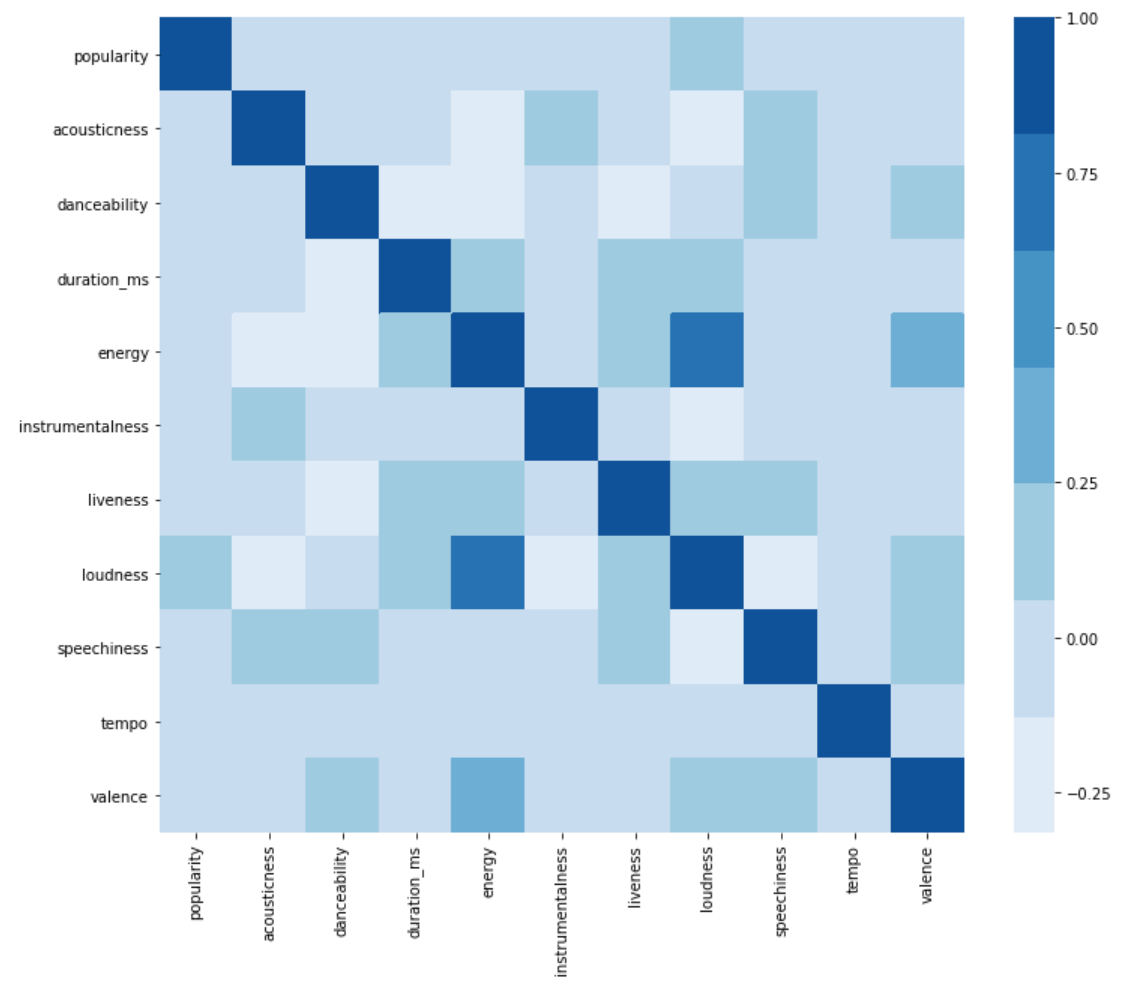
* + - 1. *Genre = Pop*

The heatmap is for Pop songs as a subset of our original dataframe. Like we can see, the only one has clear positive correlations to popularity is danceability.



* + - 1. *genre = Rap*

### This heatmap is for all Rap songs in the original dataset, as we can see that based on the graph, loudness is the feature that positively correlated to popularity



* + 1. *After Simplification*

Since the heat map only show us a vague view of how features correlated with popularity, we will not use here again. Instead we will use statsmodels [3] here to perform multilinear regression and to find the coefficients for each of the feature.

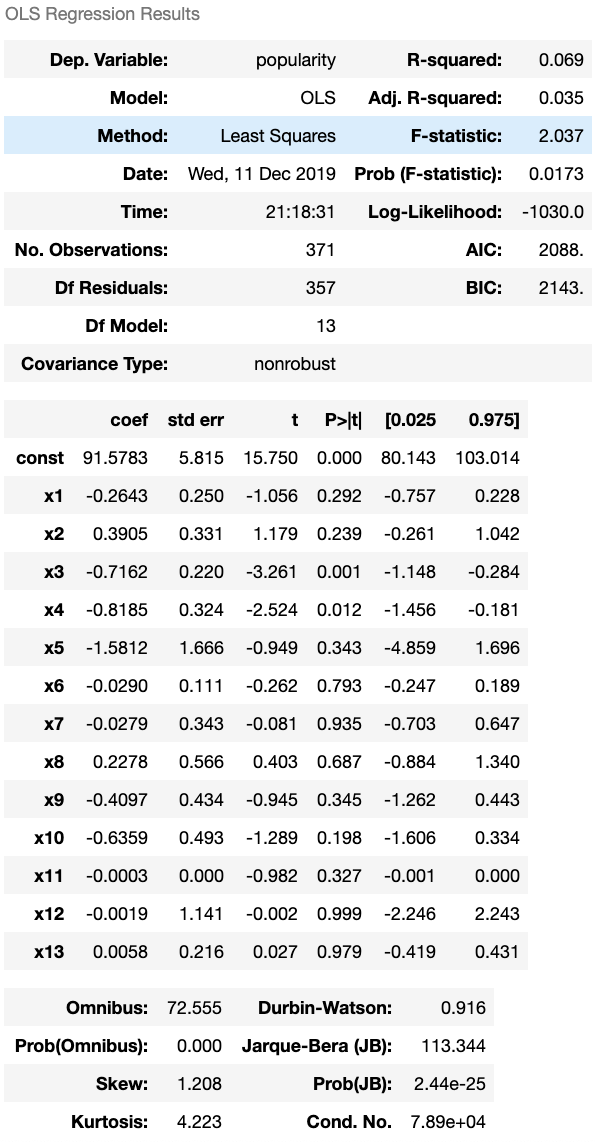
Doing the same process that takes out the subset of the dataset by genre, we will perform multilinear regression on each top genre to see what is the coefficient looks like.

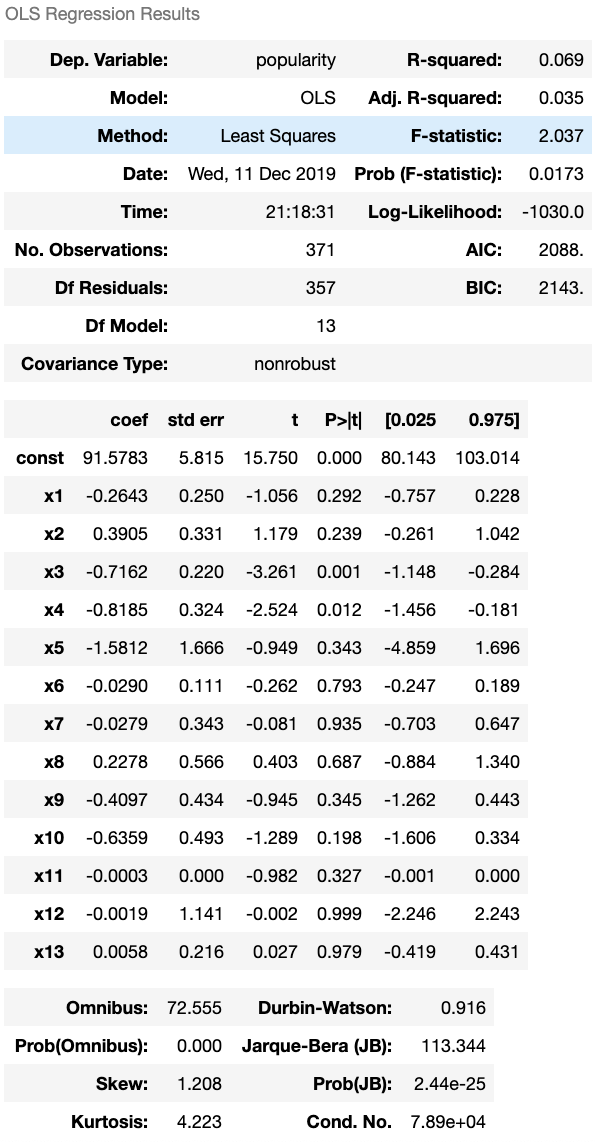
Below we have an example of the model summary, and the column will tell us about the feature correlation

* + - 1. *Genre = Pop*

Take Pop song’s model summary as an example, x1-acousticness, x2-danceability, x3-duration\_ms, x4-energy, x5-instrumentalness, x6-key, x7-liveness, x8-loudness, x9-mode, x10-speechiness, x11-time\_signature, x12-tempo, x13-valence, and y is popularity.

After multi-linear regression, the results showed us that Danceability, loudness, valence are the features positive affect popularity, which agree with the result in 5.4.1.1.





* + - 1. *Genre = Rap*

Perform the same multi-linear regression for rap, the features positively correlated are Danceability, instrumentalness, key, liveness, time\_signagure, valence.

* + - 1. *Genre = Rock*

Positively correlated features are are danceability, energy, speechiness, tempo, time\_signature, valence.

* + - 1. *Genre = Hip-Hop*

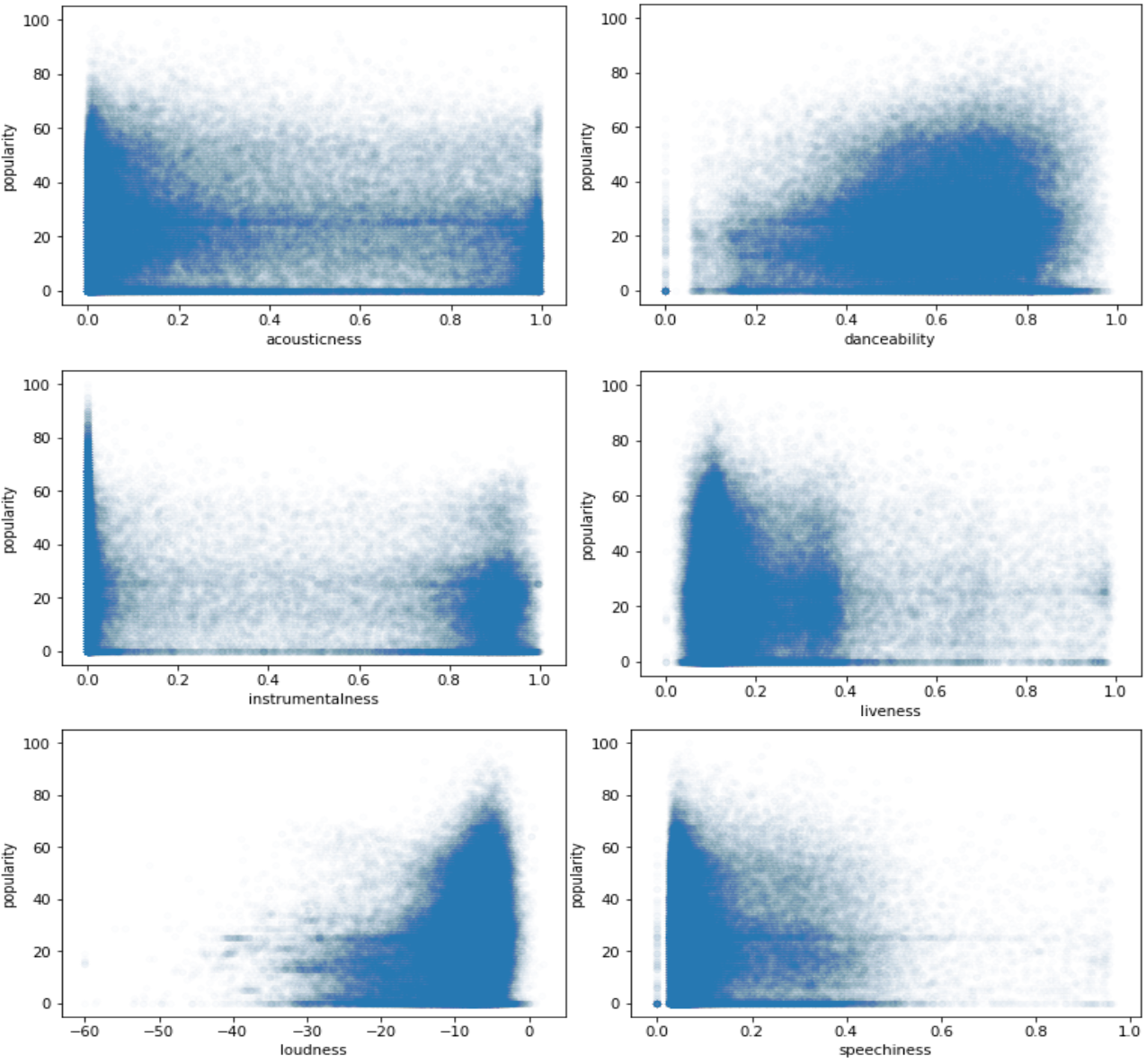
Positively correlated features are are acousticness, danceability, instrumentalness, key, livessness, tempo, time\_signature.

* + - 1. *Genre = Dance*

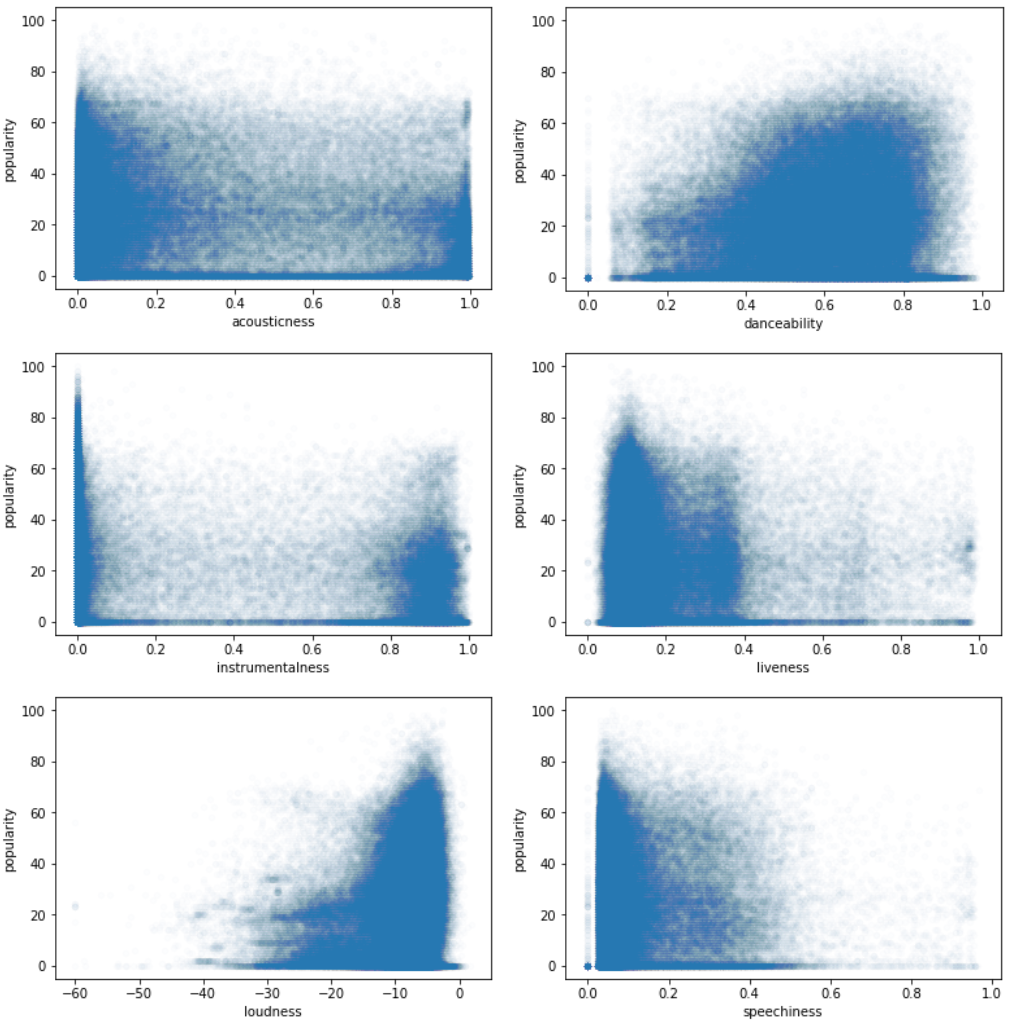
Positively correlated features are are a danceability, liveness, speechiness, time\_signature.

* 1. **Correlation Analysis**
     1. *New Datasets Analysis*

We trusted that the information in our original dataset was correct but we wanted to verify that the data we retrieved from it was somewhat consistent with other datasets. The most visual way of verifying this would be by comparing its heat maps of the features with popularity to other datasets heatmaps of features with popularity.

* + - 1. *New Dataset November 2017*

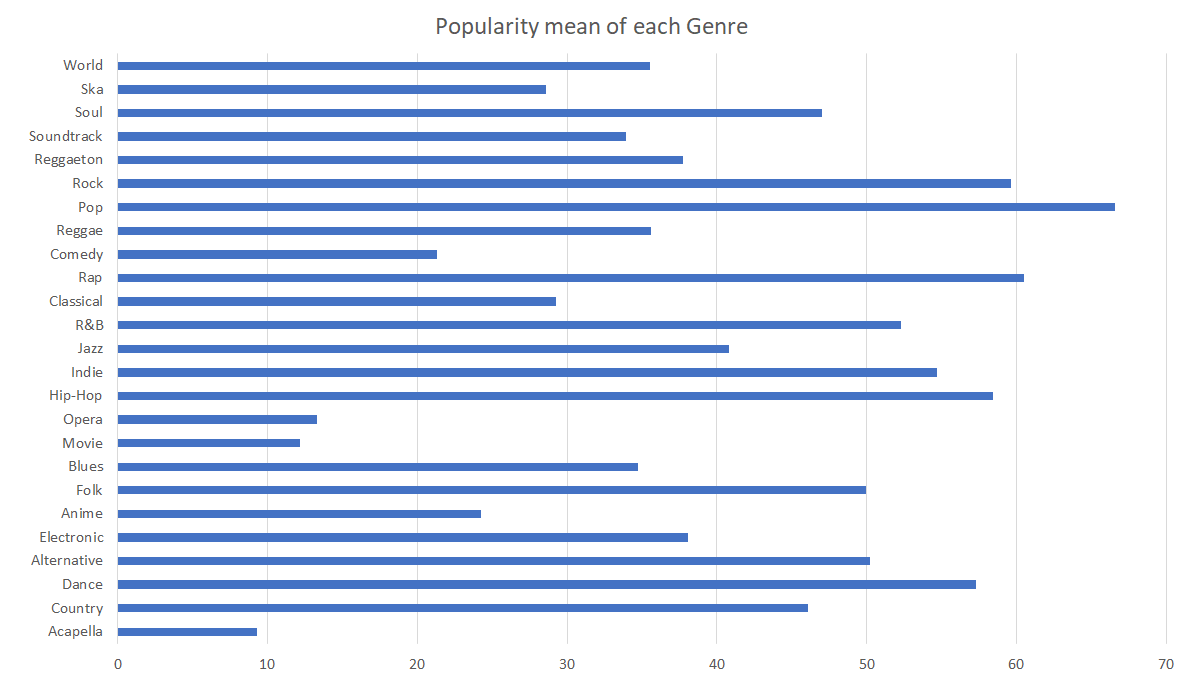
This heatmap is from a Dataset of 2017, as we can see there are pretty similar trends to our original dataset, so we can pretty much verify the legitimacy of our data. We see some shifts in the scatter plot and that's because there has been a change in the popularity trends over time.

* + - 1. *New Dataset April 2018*

As we can see, there isn't hardly any noticeable differences between this dataset and the previous one. With this we can tell for a fact that our information and data is reliable for analysis.

**Popularity by Genre**

A common question that comes up is what is the most popular music genre nowadays? We will figure it out by analyzing the popularity mean of each genre in our original database



Unsurprisingly the most popular genre is pop. The second is Rap and close third is Rock.

**Chances that a song is popular given its genre and a dominant feature of the song.**

The biggest correlations we found are the following:

-Chances a song is popular when high danceability is combined with: Rap = 99.72%, Pop = 99.62%, Rock = 99.33% ...

-Chances a song is popular when high energy is combined with:

Rap = 99.70%, Pop = 99.60%, Rock = 99.34% ...

-Chances a song is popular when high acousticness combined with: Pop = 99.82%, Rap = 99.63, Rock = 99.34% ...

-Chances a song is popular when high liveness combined with:

Rap = 100%, Rock = 99.86%, Pop = 99.52% ...

-Chances a song is popular when high instrumentalness combined with: Rap = 100%, Rock = 99.76%, Pop = 99.62% …

As we can see, besides Pop, Rap is the genre which its popularity is more directly influenced by these features.

* + 1. *Evolution of Music Trends 2016-2019*

We were able to extract very good information from this analysis. Once we obtained the top 100 songs of each year and performed analysis on the popularity of the features over the years some direction about where the music features have been heading these past few years showed up, this will help us see what have been the features that have changed the most and in which direction they have moved, if these features have become more or less popular. Thanks to this we will be able to draw better conclusions and make more accurate predictions.

* + 1. *Top Artists By Number of Hits*

|  |  |
| --- | --- |
| 2016 | 2017 |
| Zara Larsson 5  Shawn Mendes 4  J. Cole 3  Alessia Cara 3  Ariana Grande 3  The Chainsmokers 3  Rihanna 3  Justin Bieber 3  Jonas Blue 3  DJ Snake 2  George Michael 2  Sia 2  Britney Spears 2  Cheat Codes 2  Calvin Harris 2  Chuck Berry 2 | The Chainsmokers 4  Drake 3  Martin Garrix 3  Bruno Mars 2  Luis Fonsi 2  Clean Bandit 2  Maroon 5 2  DJ Khaled 2  The Weeknd 2  Calvin Harris 2  Ed Sheeran 2 |
| 2018 | 2019 |
| XXXTENTACION 6  Post Malone 5  Drake 4  Maroon 5 2  The Weeknd 2  Ariana Grande 2  Ozuna 2  Migos 2  Selena Gomez 2  Clean Bandit 2  Calvin Harris 2 | Ariana Grande 8  Why Don't We 3  Ava Max 2  Lady Gaga 2  The Chainsmokers 2  Lil Baby 2  Shawn Mendes 2  Daddy Yankee 2  Luis Fonsi 2  J. Cole 2  Bebe Rexha 2  ChocQuibTown 2  DJ Snake 2 |

As we can see there are some artists that can be seen in multiple years.

Calvin Harris: 2016, 2017, 2018

The Chainsmokers: 2016, 2017, 2019

Ariana Grande: 2016, 2018,2019

DJ Snake: 2016, 2019

Drake: 2017, 2018

Luis Fonsi: 2017,2019

Shawn Mendes: 2016, 2019

Now if we go by the total amount of hits from 2016-2019, we have some winners:

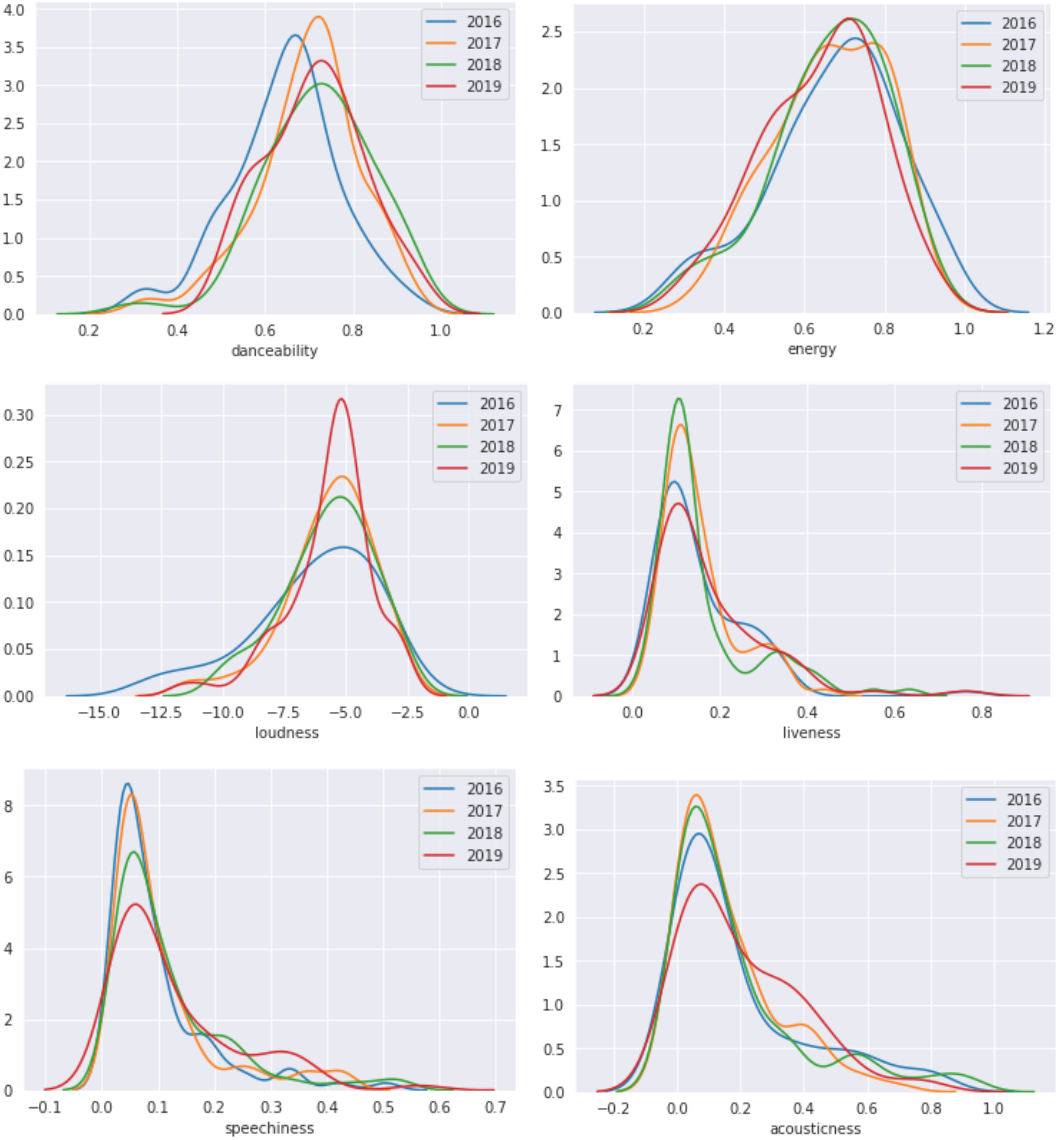
The Chainsmokers: 9

Arianna Grande: 8

Shawn Mendes: 6

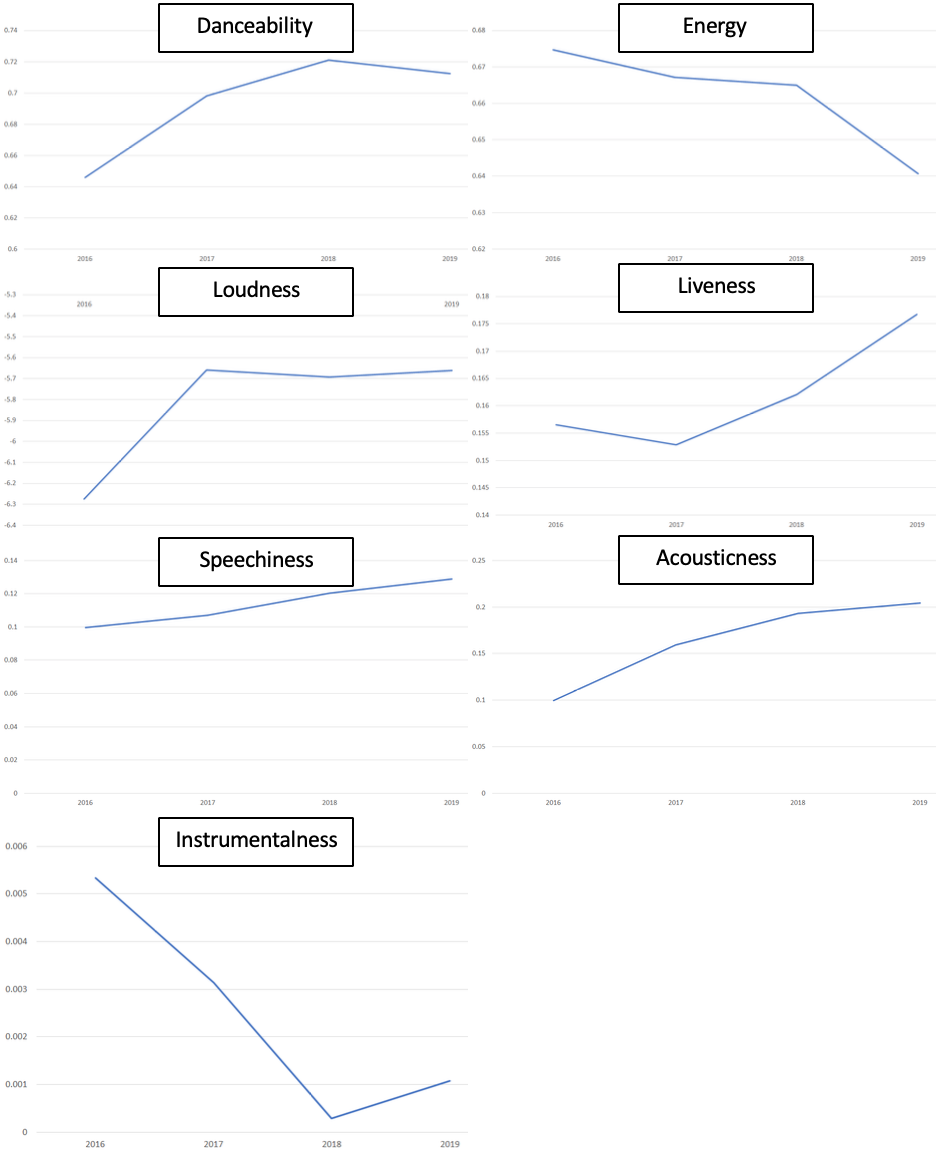
XXXTENTACION: 6

* + 1. *Plotting Features*



As we see above, in most of the graphs we can see a certain pattern of increase or decrease in the frequency of that certain feature over the years. This is most noticeable in liveness, speechiness, loudness and acousticness. Even though this gives us a visual representation of the change in the frequency of the features its still hard to tell in most cases if there is an increase or decrease in the frequency. Therefore we must take this a step further.

* + 1. *Evolution of popularity of each feature (2016-2019)*



In order to get a simpler more straightforward representation of how the popularity of the features has been changing we are going to plot the means of each feature throughout the years.

When doing this we feared that we might not find any sort of pattern, lucky enough we found some clear patterns in each one of the features throughout the years. Features that can be analyzed:

* + - 1. *Danceability (1st row left)*

It increases at a decreasing speed until 2018, then from 2018 to 2019 it decreases. We can say that popular songs have been increasing in danceability from 2016-2018 but are recently starting to become less danceable.

* + - 1. *Energy (1st row right)*

This one seems to be decreasing at different rates every year, but over all we can energetic songs have become less popular over the years, especially this past year

* + - 1. *Loudness (2nd row left)*

There is a significant increase in loudness from 2016 to 2017, since then it had a small decline but it's generally been increasing. We can say that songs have become louder over the years

* + - 1. *Liveness (2nd row right)*

This one decreased a bit at the beginning from 2016 - 2017 but after that it increased at an incredible rate. Its safe to say that songs have become livelier in the past three years.

* + - 1. *Speechiness (3rd row left)*

Speechiness is pretty straight forward, it pretty much has been increasing at a constant rate since 2016.

* + - 1. *Acousticness (3rd row right)*

This one has been increasing since 2016 but at a decreasing rate.

* + - 1. *Instrumentalness(4th row left)*

As we can see, instrumentalness decreases at an incredible rate from 2016 till 2018, we see it go up slightly from 2018 to 2019

* + 1. *Future Prediction*

Now that we have how the popularity of each feature has been changing over time, we want to estimate how that feature is going to change in the near future, is it going to increase or decrease? Here we calculate how much the feature has increased or decreased from 2016 to 2019, and if it's currently increasing or decreasing.

Danceability: + 10.33% → Decreasing

Energy: - 5.04% → Decreasing

Loudness: + 9.76% → Increasing

Liveness: + 12.91% → Increasing

Speechiness: + 29.42% → Increasing

Acousticness: + 105.12% → Increasing

Instrumentalness: - 79.89% → Decreasing

It’s also important for future predictions to emphasize on what the features look like right now. This is the mean value of each feature in 2019: Danceability: 0.71, Energy: 0.64, Loudness: -5.66, Liveness: 0.17, Speechiness: 0.12, Acousticness: 0.2, Instrumentalness: 0.00107.

This information is key to have a better estimate on what the trends are going to be in the near future, and the information can be used by artists and music producers to get a good idea on what type of songs they could make if they want higher chances of the song becoming popular. It can also be used by anyone interested in music to see what the near future music is going to sound like.

* + - 1. *Most Likely Increase*

We can predict that loudness, liveness, speechiness and acousticness are the safest bet of music features increasing in the near future, that's because they have mostly been increasing since 2016 and were still increasing in 2019.

* + - 1. *Most Likely Decrease*

The most obvious to decrease is energy given its history. The other one most likely to decrease is danceability, which even though has increased a bit since 2016 it has been increasing at a decreasing rate, and this last year it decreased.

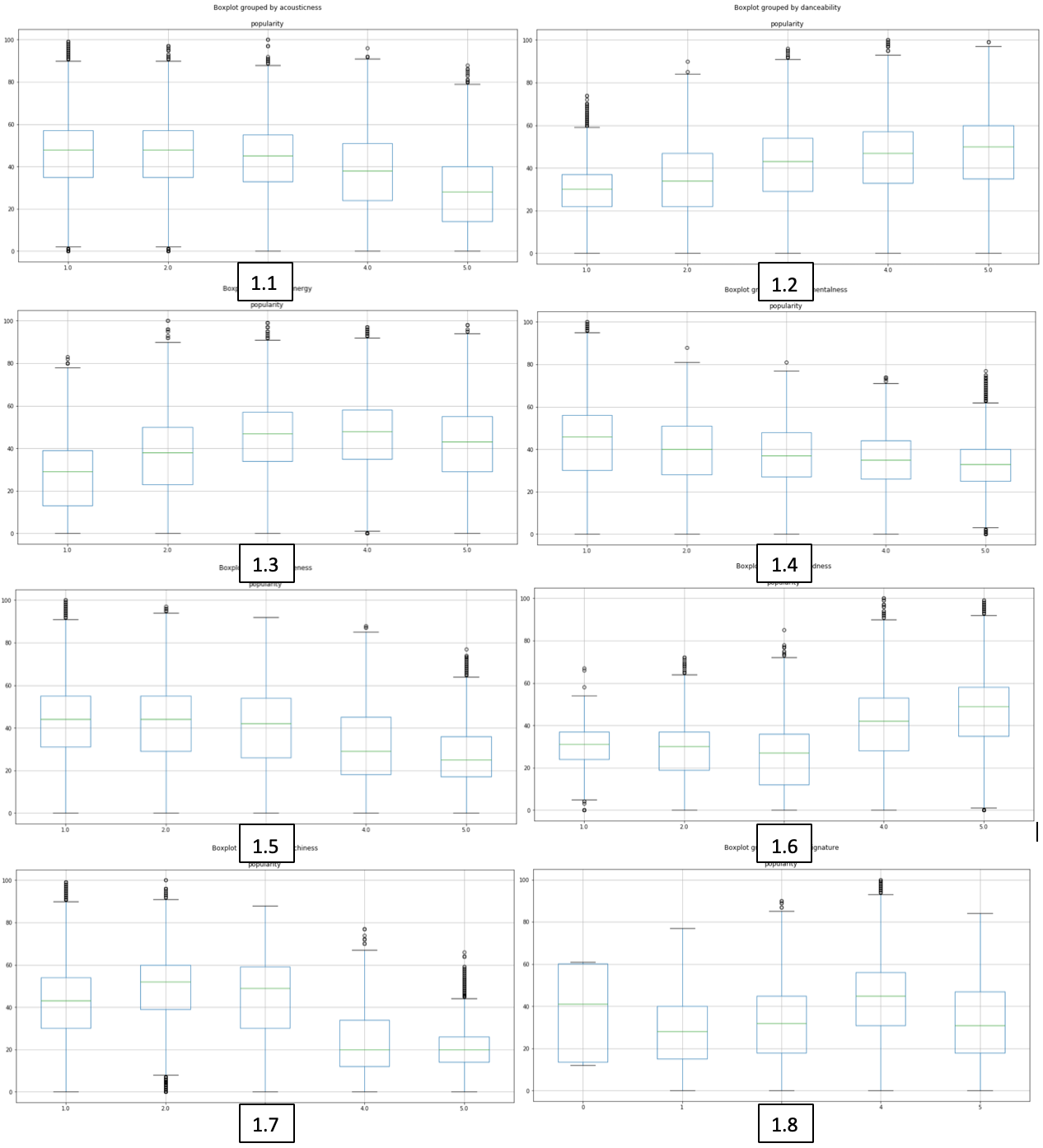
* + 1. *Hard to Predict*

Instrumentalness has increased this past year but has a record of an incredible decrease from 2016 to 2019, so it’s hard to say how long this increase will actually last.

* 1. **Simplified Data Features Results**

After group each features into 5-6, convert nominal data to numerical data also in groups, we are able to simply use boxplots to see what are the characteristics for different groups in the same feature. And it turned out to be significant

* Acousticness (1.1): lower score, higher popularity (0-0.2);
* Danceability (1.2): higher score, higher popularity (0.8-1.0);
* Energy(1.3): higher score, higher popularity, but peaked at group (0.6-0.8);
* Instrumentaness(1.4): lower score, high popularity(0-0.2)
* Liveness(1.5): lower score group has higher popularity (0.-0.4)
* Loudness(1.6): higher score, higher popularity (0.8-1.0), with the lowest (0.4-0.6)
* Speechiness(1.7): mid groups (0.2-0.6) has higher popularity
* Time\_signature(1.8): mid group (0.6-0.8) has the highest mean popularity



* 1. **Simplified Data Linear Logistic regression**

Initialize the weight and use the 80% simplified data to train and test on the remaining 20% of the data. Based on the simplited datasets. Removing the track\_name and track\_id because it has too many distinct values.

After trained, the trained weight as been applied to test set and average, the correctly prediction percentage is about 65%.

* 1. **Logistic Regression Model**

We developed a Logistic Regression using split of 80/20 for the training and test set respectively. We also did recursive feature elimination to find the features that gave the best results which in this case were : “'acousticness', 'danceability', 'instrumentalness', 'speechiness' and 'valence'. Also we modified the popularity feature to have a value equal to 1 if the popularity is more than 35 and 0 otherwise. We selected the value 35 because 70% of the songs are below that number and this way we have balanced data.

The accuracy of our classifier on the test set was 68% accuracy on average which this is is a significant improvement from the linear regression because taking into account that our test set has 46.000 songs a 3 % improvement is significantly better.



* 1. **SGD Model**

We developed a Stochastic Gradient Descent using split of 80/20 for the training and test set respectively. We also did recursive feature elimination to find the features that gave the best results which in this case were : 'energy', 'instrumentalness', 'liveness', 'speechiness', 'valence'. We did the same modification to the popularity feature for this model.

The accuracy of our classifier on the test set was 69% accuracy on average which this is is a significant improvement from the linear regression and a little improvement from the logistic regression.

.

**FUTURE WORK**

In the future, an even more broad dataset we can get will be awesome. Not only use the existing dataset we found online, we will use Spotify api to retrieve our own songs and features expand it to a longer expansion of time.

Lyrics analysis will be super useful as well, but it requires Natural Language Process to find potentially patterns and train to have a model to be able to take account of the lyrics. Another aspect of lyrics analysis is that it might be closely related to historical data or hot event happening at the time the song is published, so if we are able to relate the historical events, or period of time with the song/lyrics, it will be even better.

Even though we have the popularity of each feature from 2016 to 2019 it would be interesting to find the popularity of each feature during different times of the year (summer, fall, winter and spring) and see if there is any correlation between the time of the year and the kind of music that is being listened.

For future predictions it would also be helpful to have more datasets with the same features we had in our original dataset. That way we can more precisely track how the music is evolving over time and analyze the trends with more precision.

# **CONCLUSION**

Different genre songs have some different features that positively correlated to popularity. They are some features are common for all the genres like danceability, liveness, time\_signature, valence. So if people are producing a song in a specific genre, we can give specific suggestions. A Pop song for example, if a company, band, song writer is going to produce a new one, a high danceability, loud, and good valence (positive, good feeling) will be more successful.

We can say that the most popular genres besides whats considered as Pop are Rap and Rock. By the analysis we have done of popular trends over the years we can say that what would help a song be popular if its higher than usual in any of the following features: loudness, liveness, speechiness, acousticness. And is lower than usual in the following features: danceability and energy.

# **REFERENCES**

[1] Spotify Tracks DB, Music database (232k tracks) key, mode and time signature are cleaned. <https://www.kaggle.com/zaheenhamidani/ultimate-spotify-tracks-db>

[2] SpotifyAudioFeaturesNov2018, Music database (116k tracks)

https://www.kaggle.com/tomigelo/spotify-audio-features

[3] SpotifyAudioFeaturesApril2019, Music database (130k tracks)

https://www.kaggle.com/tomigelo/spotify-audio-features

[4] Top2017, Music database (100 tracks)

https://www.kaggle.com/nadintamer/top-tracks-of-2017

[5] Top2018, Music database (100 tracks)

https://www.kaggle.com/nadintamer/top-spotify-tracks-of-2018

[6]Jupyter Notebook, <https://jupyter.org/>

[7] [pandas.DataFrame - pandas 0.25.3, documentation](https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html)<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>

[8] StatsModels, Statistics in Python, <https://www.statsmodels.org/stable/index.html>

[9] Scipy.org <https://www.statsmodels.org/stable/index.html>

[10] Seaborn: Statistical Data Visualization <https://seaborn.pydata.org/>

[11] All Models, Angle-based Outlier Detector (ABOD) <https://pyod.readthedocs.io/en/latest/pyod.models.html>

[12] Matplotlib, <https://matplotlib.org/>

[13] Spotify Data Project. <https://github.com/tgel0/spotify-data>

[14] Spotify For Developers, <https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/>

[15] regional-global-weekly-2016-12-23--2016-12-30.csv (renamed to top2016), Music database (200 tracks)

<https://spotifycharts.com/regional/global/weekly/2016-12-23--2016-12-30>

[16] regional-global-weekly-2019-03-22--2019-03-29.csv (renamed to top2019), Music database (200 tracks)

<https://spotifycharts.com/regional/global/weekly/2019-03-22--2019-03-29>

[17] Building a logistic regression, <https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8>

[18] Scikit-learn, <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html>

# **APPENDIX**

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. All the work has been done by individual group members.