Predicting Future Popular Music and a System of Recommending Music

Group 5

Introduction

Over the past decade, online streaming services such as Apple Music and Spotify have become major suppliers of music to huge audiences. Due to their popularity these streaming services provide a plethora of data that can be used by record companies to inform and advise on business decisions such as which artists to sign, which audiences to target and how to build a positive reputation among customers and artists.

Music bulletin boards from these services, used alongside information from the charts (lists of the best selling and most streamed albums and tracks, in the UK) are becoming important indicators for suggesting which artists have been (or still are) popular.

The key to maximizing profit is to dedicate resources and capital onto artists whose music is popular. Being capable of predicting popular music types is therefore essential to the growth of the company as they will be able to sign more artists who are likely to become popular (based on the type of music they produce).

As well as being able to sign artists that are likely to be popular it is also important to be able to attract more customers. Recommending new music based on a current song being listened to is a key component of streaming services that many users often look for. An accurate system like this could be very valuable for a record company should they look to develop a streaming system for their music.

The Problem

We need to be able to accurately predict trends within the music industry to try and stay on top of our investments into artists. This is key to ensuring we invest in the best artists in order to maximise return from investments and overall profits, something that is vital for success as a company. We need to define what popularity means within the context of the music industry and be able to quantify this in order to find what track features make a song more or less popular. Based on this, we can then find what music is the most beneficial to invest into.

Customers listen to varying kinds of music and as a company it would be useful to be able to recommend music for a user based on the music they have been listening to. It is vital that this system is consistent and can provide sensible suggestions as this will help in attracting and keeping customers, building a reputation for the company as technically advanced and reliable.

Data

To complete the task and test the quality of the developed systems, a subset of Spotify's database was used alongside data from the Official Charts Company. The Spotify dataset (Everitt, 2019) is composed of tracks from albums which were chosen manually with some degree of arbitrariness. We worked with 35 albums from each decade from the 1960's right up to the 2010's. Spotify provided data on each track to describe the key attributes (e.g. speechiness, valence, etc). These are all contained within category four, defined below. Further details can be found on the Spotify Web Api Guidance page (Spotify Developers, 2019).

We also have features from the Official Charts Company such as the number of weeks on the chart, the top chart position reached by an album and the number of weeks spent at number one. These variables predominantly give information relating to the first task in terms of quantifying success.

Each track in the database has 28 variables relating to it. These variables can be separated into four categories:

1. **Artist Identity** - Name, ID, Genres, Popularity, Number of Followers
2. **Album Identity** - Name, ID, Best Chart Position, Nuber of Weeks on the Charts, Number of Weeks at Number One, Popularity, Release Date
3. **Track Identity** - Name, ID, Duration, Number
4. **Track Technical Characteristics** - Danceability, Energy, Key, Loudness, Mode, Speechiness, Acousticness, Instrumentalness, Liveness, Valance, Tempo, Time Signature

The Plan

Predicting future popular music is a relatively tricky concept as popularity is not well-defined and subjective to individual tastes. Within the data set there are multiple indicators that could potentially quantify popularity. Initially we considered included "AlbumWeeksOnChart", "ArtistPopularity" and "AlbumPopularity". Due to the fact that some very popular albums (based on "AlbumPopularity" and "ArtistPopularity") never actually entered the charts. The Kinks Are The Village Green Preservation Society by The Kinks never made it onto the charts despite them having a very high "ArtistPopularity" score and the album being described as "the most successful ever flop" (Mojo Staff, 2018) and reaching 258 on Rolling Stone magazine's list of the 500 greatest albums of all time.

We realised "AlbumWeeksOnChart" is a highly unpredictable and inconsistent attribute as an indicator for popularity. This would result in the gambling of capital and resources on something that is almost random, so it was not considered. "ArtistNumFollowers" was then looked at as an alternative option and selected to be considered in more depth in the analysis section below due to it being more consistent and positively correlated with "ArtistPopularity" and "AlbumPopularity". The goal was to associate these three quantifiers for popularity with predictors from the rest of the data set in the form of a linear model in order to suggest which artists may be popular.

We interpreted the process of producing a system for recommending new music based on taste to be predicting music similar to what the listener is currently listening to. This raised the question of what kind of music is considered similar. Initially it was thought we could pick out the most significant characteristics of a track and match these with new tracks that have similar values in these characteristics. We soon realized this is not an accurate way to predict since you could end up with very inaccurate suggestions. For example, a heavy metal track and one categorised under vibrant symphony could both have low speechiness, high tempo, high loudness. It would be unreasonable to recommend a symphony to someone who is listening to heavy metal music. If we would like to be more accurate, then it would require the inclusion of many more descriptors to match and with so many available characteristics for each track and the number of tracks to search through it would simply be too time-consuming.

Alternatively genres could be associated with certain track characteristics. The track played will be associated with a genre based on the values of its descriptors, then another track categorised under that genre will be recommended. However, there are almost 250 genres in the data set and many of them are very similar. Genres often have a lot of overlap (e.g. there are many varieties of rock), making it difficult to classify the tracks accurately into one genre. The best way to handle this will be investigated further in the analysis stage.

Analysis

Exploratory Data Analysis (EDA)

Task 1

For all analysis carried out for this first task, individual track information was removed and each album was labelled with the average characteristics of tracks on that album. For example, "AlbumLiveness" would be the average "TrackLiveness" score for all tracks on a particular album. This was done because all popularity variables relate to either an artist or an album, never an individual track.

It's reasonable to suggest tastes of listeners and what types of music are popular will change over time. To get an idea of how much things changed we can look at the following two graphs which look at either end of the time period our data is within. Only the genres that appear more than three times in the database are included in these graphs in order to avoid plotting too many. It is worth noting, most albums have multiple genres associated with them so they may be counted multiple times for different categories.

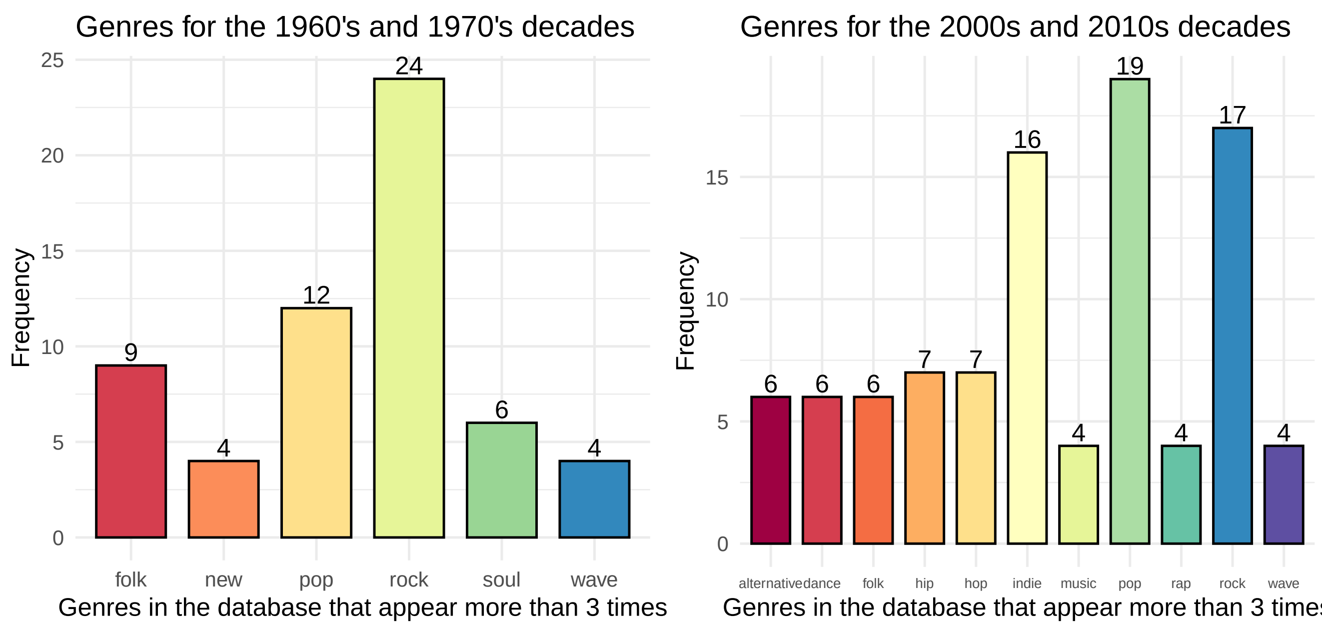

Figure 1: Comparison between most common genres from 1960-1979 and most common from 2000-2019

From these it is clear that the types of music people are interested in has changed over the years. "Pop" and "rock" have stayed at the top but recently, "indie" music for example has become much more popular. Therefore, for predicting future popular music types, we are only using data from the 2000s and 2010s, the most recent decades (this applies to the rest of the EDA carried out for the first task as well). Only using two decades worth of albums does limit the data available to us quite considerably, but it should still provide more accurate predictions for the present day, which is what we want (we don't need to be able to predict if an artist would be popular in the 1960s, we want to know who will be popular now).

We must also look at the different characteristics for tracks released in the past 20 years and how they relate to the three variables suggested in our initial plan; ArtistPopularity, AlbumPopularity and ArtistNumFollowers. For an overview of these we can look at a correlation coefficient table, shown here:

	Artist Popularity	Artist Followers	Album Popularity
Acousticness	-0.098	-0.099	-0.069
Valence	-0.185	-0.055	-0.206
Energy	0.009	0.019	0.006
Danceability	0.040	0.040	-0.007
Loudness	0.061	0.089	-0.003
Speechiness	0.155	0.145	0.237
Instrumentalness	-0.050	-0.166	0.040
Liveness	-0.004	0.103	0.120
Tempo	-0.153	-0.066	-0.131

Table 1: Correlation coefficient table for popularity quantifiers against track characteristics

From this table we can see that characteristics such as valence, speechiness and tempo may be useful to look at when fitting a model for predicting "ArtistPopularity" since there are stronger correlations for these under "ArtistPopularity" than there are for others. The correlations can be visualised nicely through scatter plots.

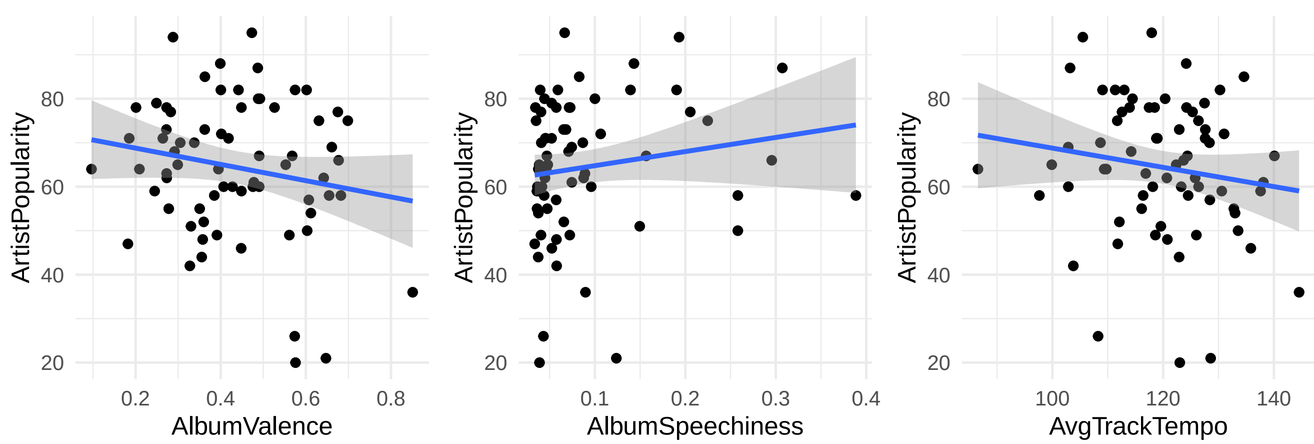

Figure 2: Scatterplots for ArtistPopularity against some track characteristics

The same method applies to looking for useful variables relating to the other two popularity quantifiers.

Task 2

In preparation for the second task, it was necessary to see which track characteristics were linked in order to be able to accurately recommend similar tracks. A principal component analysis (PCA) was performed on a group of nine key track characteristics, with the goal of identifying collinear or associated characteristics and being able to group them into clusters. These clusters can then be used to prioritise certain characteristics when recommending a new song. Only nine of the track characteristics were included as we deemed discrete variables such "TrackKey", "TrackMode" and "TrackTimeSignature" to be less significant due to the fact they can only take a very small range of values. For further details on this EDA, see section 1.1.2 and 1.1.3 of the technical appendix.

Predicting Future Popular Music (Task 1)

From the graphs above, whilst there are clearly some correlation coefficients much stronger than others, there are none that could be considered very strong (i.e. almost 1 or -1). In a situation like this, where our predictors (track characteristics) are not particularly linearly related to the dependent variables (our three popularity quantifiers), fitting a generalized linear model is often a good idea. These provide useful linear equations for predicting new values (in this case, predicting popularity values) and are relatively straightforward to test goodness of fit by using Chi-squared tests and F-tests.

The same method of fitting the model was used for all three popularity variables. Here we shall briefly look at how the model for "ArtistNumFollowers" was created, for more detail on the fitting of these models and how each is used, see section 2.3 of the technical appendix.

Any track characteristic with a correlation coefficient between itself and "ArtistNumFollowers" of ≥ 0.05 was included in the initial fitted model. At first this gave us quite a few predictors; "AlbumSpeechiness", "AlbumLoudness", "AlbumValence", "AvgTrackTempo", "AlbumAcousticness", "AlbumLiveness" and "AlbumInstrumentalness".

By completing a Chi-squared test (for goodness of fit - how far observed values stray from the expected values) and an F test (for significance of regression - comparing our defined model to one with no predictors at all) we can get an idea of which predictors are not as significant from their p-values. The smaller the p-value for each predictor, the better, with those ≤ 0.05 considered significant (in this case, optimal).

Using this method, eliminating predictor variables with the largest p-values at each step, we eventually reach a model where all predictors have significant p-values, implying these are the optimal choice for this model.

In the case of "ArtistNumFollowers", after the first test, we had the following results.

	Chi-sq p values	F-test p values
Album Speechiness	0.216	0.287
Album Valence	0.010	0.029
Album Loudness	0.596	0.647
AvgTrack Tempo	0.554	0.609
Album Acousticness	0.033	0.068
Album Instrumentalness	0.242	0.313
Album Liveness	0.588	0.640

Table 2: Chi-squared and F test p values on the initial fitted model

From this, we dropped AlbumLoudness, AlbumLiveness and AvgTrackTempo and repeated the test again with the remaining predictors. After doing this a couple of times, we are left with only three predictor variables that have significant p-values; "AlbumValence", "AlbumSpeechiness" and "AlbumAcousticness", providing us with the following model.

$$ArtistNumFollowers = \alpha + \beta_1 AlbumValence + \beta_2 AlbumSpeechiness + \beta_3 AlbumAcousticness$$

Conveniently, when all three models are fitted it happens that the three characteristics best for predicting "ArtistNumFollowers" also provide the most accurate predictions for "ArtistPopularity" and "AlbumPopularity".

$$ArtistPopularity = \alpha + \beta_1 AlbumValence + \beta_2 AlbumSpeechiness + \beta_3 AlbumAcousticness$$

$$AlbumPopularity = \alpha + \beta_1 AlbumValence + \beta_2 AlbumSpeechiness + \beta_3 AlbumAcousticness$$

The coefficients for each, shown below, are also all of the same sign. This means for each model, an increase in speechiness will likely increase popularity whereas an increase in valence or acousticness will have a negative effect on popularity.

$ArtistNumFollowers$	$ArtistPopularity$	$AlbumPopularity$
$\alpha = 16.93$	$\alpha = 4.46$	$\alpha = 4.43$
$\beta_1 = -4.31$	$\beta_1 = -0.66$	$\beta_1 = -0.99$
$\beta_2 = 6.52$	$\beta_2 = 0.95$	$\beta_2 = 1.66$
$\beta_3 = -2.83$	$\beta_3 = -0.27$	$\beta_3 = -0.36$

These models, with the given coefficients, can now be used to predict how many followers an artist might have, what their popularity score may be or how popular their albums will be based on their style of music. It is reasonable to recommend investing in artists that make music with a lower valence rating, low acousticness and a higher speechiness score to have the best chance at being popular and making a profit.

Testing

To test our models it only makes sense to test data from the two decades which we built the model on; the 2000s and 2010s. Only four albums were used in the testing as these were the only ones from the relevant decades in our given test set with the mean absolute error being used as a measurement of accuracy. Albums used for testing include "The Streets: Original Pirate Material", "Missy Elliott: Miss E...So Addictive", "Public Service Broadcasting: The Race For Space" and "Daft Punk: Random Access Memories". The average error (difference between predicted and actual values) for each model is as follows (for full details see 2.4.2 of the appendix):

	Artist Followers	Artist Popularity	Album Popularity
MAE error	2458391	14.65	12.26

Table 3: Mean absolute error for the models tested on four test albums

Limitations

- The database for the decades 2000s and 2010s is very limited. Given more data from these two decades the model could be a lot more accurate (or use different predictors).
- The tests we conducted on the models could be improved. For example, AvgTrackTempo has a high correlation with the three variables we defined as quantifying popularity but we didn't include it due to the results of the Chi-squared and F tests conducted.
- Trends of popularity may shift just like in the past 50 years. Therefore, this model should be run periodically when new data becomes available to make regular adjustments, accounting for new trends.

Song Recommendation System (Task 2)

Song recommendation is a tricky process. You can either just recommend a song from the same genre, or recommend a song with similar characteristics. This leads to the question of which one is better at the overall recommendations? Each has its flaws and it's hard to use both when trying to recommend. For instance, if you used solely the characteristics of the song like the valence, danceability and so on we can have songs with very similar characteristics falling into different genres. An example is, if you have a song with high valence and high energy this could fall into either the hard rock or pop genre, meaning we could end up with a poor recommendation. Using solely the genre, Spotify has so many, seemingly arbitrarily assigned genres, that recommending this way could lead to very few songs/albums to recommend.

Via grouping the songs in albums, and averaging their technical characteristics, we can find albums of similar characteristics to recommend songs from them to the user. This has led to us using the characteristics in a grouped manner, meaning if we use all 9 of the main descriptive characteristics used in the EDA, we should end up with fairly accurate predictions. At the same time, by comparing "album to album" and not "song-song", we are giving the user a variety of songs to listen to rather than just one. This will account for albums that may have one song drastically different from the rest, for example if a pop artist decided to record one heavy rock song then the recommendation system would still be able to suggest similar pop songs as well as the outlier in the rock song.

The system works as a supervised learning problem. We are providing the system with the album of the current track. We are then aiming to find out which group of albums this album belongs to based on its average track characteristics. We group the albums through a k-means clustering, the details of which can be seen in section 1.1.3 and the application of it within our function in section 3.3 of the technical appendix.

Before this can be done, the database being used must be grouped by album, with each one being labelled with the average value of the technical characteristics of its songs. We must then check if an album is already in the current database. If it is not, then the album is added into the database. If the data of the album to be added is not of the same format as the current database, an error message is displayed.

Within our recommendation system we have a "predict" function. This groups the albums in the database into clusters of similar albums based on their technical characteristics. This is done by the k-means clustering method mentioned above. Once this is done, we will select an album randomly from the same cluster (ensuring the inputted album is not returned again by removing it from this cluster before selection) as these should contain tracks of a similar type to the one currently being listened to.

If we recommend an album to the user they may not know which song to start with, so we then randomly select one track in that album to recommend a specific song for the user to start with. This way we have given them a good selection of songs to listen to with a specific recommendation with which one to start with.

Demonstration

It is of course impossible to test if a song recommendation is correct as it is simply a recommendation and so some users may love it where other users may not like it at all. The only way we can look at this is visually through a demonstration. Here we will assume a user is listening to the album Led Zeppelin by Led Zeppelin and the recommendation is:

```
## [1] "Your album recommendation: The Proclaimers - Life With You"
## [1] "Your song recommendation: The Long Haul"
```

The way the code is run is shown in the appendix sections 3.1-3.3 with more examples.

Limitations:

- If the album is not already in the database and does not have all of the same attributes as the Spotify database, then the system will not work. For databases with a different structure, the system must be altered so it can retrieve the characteristics of tracks in the new database.
- The database is very limited: if an album is entered that has nothing similar to it in the database then the recommendation will be not suitable (but will just have similar technical characteristics).

An example of this from the data set used for this project is any album in the classical genre. We do not have many (if any) albums from this genre so the recommendation may not be very suitable.

- The number of clusters created is arbitrarily set to 20. If a way of choosing the clusters inside the code to optimise this decision could be implemented it will improve the system dramatically.
- The code is programmed using album characteristics and recommendations. This is due to the test data provided being albums but usually the user will listen to a specific song, not a specific album.
- We haven't used the genre of the song at all when producing the recommendation. Thus, implementing a way to include this could improve the suitability of the recommendation.

Conclusion

As shown through the tests and demonstrations carried out above, both of these system implementations provide good, statistically sound results. It is clear though that both still have their limitations and room for improvement.

Based on our models, we have been able to conclude that artists producing tracks that have a higher speechiness rating and a rating for valence and acousticness on the lower end of the scale will likely be more successful. From the second task, we are now capable of suggesting new music to a listener based on what they are currently listening to. Unlike the popularity prediction system, this recommendation system is applicable to whichever database it is given. For predicting popularity, if a new, better and more extensive database is provided then the models will need to be re-fit based on the new data.

Accuracy is undoubtedly limited due to the relatively small size of the database being worked with and notable lack of any albums of certain genres such as electronic or classical music. Given a much more extensive database, the models fitted for predicting popularity in task one could change considerably. For the song recommendation system, with more data available, the albums could be grouped into a much larger number of clusters (we currently use 20 as stated in the limitations of this section). This would allow for the the next suggestion to be selected from a much more closely related group of albums.

Given more time and a larger database, alternative methods could be investigated such as implementing a "k nearest neighbours" algorithm to predict the popularity of an album or artist based on those albums or artists with the most closely related average track characteristics. For now, with the available resources, the systems produced provide sufficiently accurate results.

References

1) Spotify Developers. (2019). Audio Features for a Track. Retrieved from Spotify: https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/
2) Richard Everitt, Simon Spencer. (2019, November). ST344:Professional Practice of Data Analysis. Retrieved from Wariwick Univeristy Moodle: https://moodle.warwick.ac.uk/mod/resource/view.php?id=836117
3) Mojo Staff. (2018). MOJO 300. In M. Staff, Mojo Magazine (p. November). Bauer Media Group.