ST344 Group 6 Report

Benjamin Charnock, Jakub Domasik, Thomas Holgate, Matthew Persin, Ziru Zhou

November 2019

1 Introduction

1.1 Data Explanation

The dataset we looked at came directly from Spotify's records, taken on the 25th September 2019. It contains a variety of albums, ranging from the 1960's to the modern day, and with this comes a wide range of genres. The dataset has the basic information about songs such as artist, album name, duration and genre, as well as a mix of information in the form of different variables such as track tempo, speechiness and valence. Comprehensive information on each variable incorporated in the dataset can be found in the technical appendix and in the Spotify API[1].

1.2 Data Cleaning

1.2.1 Album Release Dates

The *AlbumReleaseDate* variable was missing a number of data points. However this was limited to only a few albums and hence we decided to complete the dataset by sourcing the release dates manually, this was done through online research. Not only were there complete dates missing but there was also inconsistency in the level of accuracy that was documented. Some albums were recorded just by year, some by month and others by exact date. In order to keep this consistent we decided to fill in the month for all the dates, so every album in our dataset had at least a month and a year of release. With this we planned to carry out seasonal analysis of the tracks and how popularity, amongst other variables, may be dependent on the time of year an album was released. Note that more albums were released in January, however research suggested that if the release month was not available, Spotify had marked these albums as released on the 1st of January for the corresponding year.

1.2.2 Genres

In the dataset albums had been assigned large numbers of very specific genres. There were 209 unique genre instances, with individual tracks having been assigned up to 19. To establish sufficiently large genres for useful analysis we opted to aggregate the Spotify genres into super-genres based off of shared words in the title. For example *bubblegum pop* and *Britpop* were aggregated into a pop variable. These super-genres were stored as binary indicator variables associated with the albums. The selected super-genres are outlined below:

- Pop
- Rock
- Techno
- Jazz
- Blues
- Folk
- Indie
- Rap
- Metal
- Punk

These super-genres were selected through a combination of research into major genres and an examination of the dataset to reveal which were common. There was no limitation on super-genre membership and albums often belonged to multiple classes. Eight albums did not fall into any genre column. However these

included albums such as "The Elder Scrolls IV: Oblivion: Original Game Soundtrack" which justifiably wouldn't fall into any conventional super-genre and also would likely not be selected for its commercial popularity as a standalone album.

1.2.3 Album Aggregation

Popularity data was provided relating either to albums or to artists, rather than to individual tracks. This made aggregation over albums necessary. The mean values of quantitative track data was calculated and combined with album and artist data to create a dataset containing albums, rather than tracks. This cleaning was for the purposes of task 1, task 2 made use of track data.

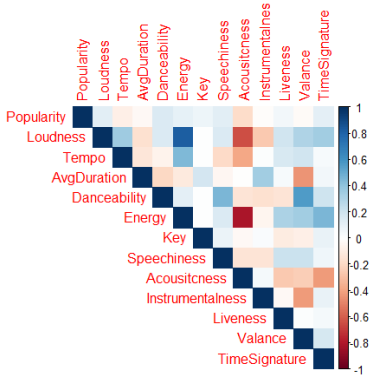
2 Exploratory Data Analysis (EDA)

The main variables to analyse for Task 1 were the popularity related variables. These included *ArtistPopularity*, *AlbumPopularity*, *ArtistNumFollowers* and *AlbumWeeksOnChart*. Album and artist popularity were variables produced by Spotify scaling their current popularity from 0 to 100. An analysis of the relationship between *ArtistNumFollowers* and *ArtistPopularity* suggested that the relationship between them was exponential. This would likely imply that a similar relationship exists between *AlbumPopularity* and the number of listeners, as the popularity metrics were synthetic.

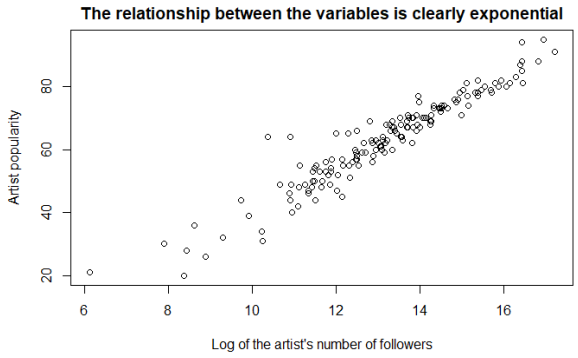
We considered a synthetic variable that took into account others such as *AlbumWeeksInChart*, *ArtistPopularity* and *AlbumWeeksNumberOne*, however this was excluded as it was biased towards more recent music, as an album that was released a year ago could have only charted for at most 52 weeks. Since a large number of albums charted for very long periods this represented a major issue favouring older music. An Album's length on the charts correlated with its release year. We discovered, when removing more recent years, this correlation was eliminated. This is likely due to more recent albums having had less time to be on the charts.

AlbumPopularity is weighted towards more recent songs[1], so recent listens count more when calculating popularity, which clearly has a bias towards new releases. However this is acceptable because the research is being conducted for the purpose of a commercial release. The mainstream popularity of current releases, which *AlbumPopularity* measures, is the area of interest for this project.

We created a correlation heat map to visualise the correlation matrix. This gave clear indications of which variables were related, and how strongly they were so. Interesting variables were then analysed in further plots, along with manual research into the Spotify IDE[1], to reveal if those relationships may be due to measurement choices.



(a) Correlation heatmap.



(b) Relationship between popularity variable and number of followers

We identified several methods of EDA as dead ends after initial analysis. A Principle Component Analysis (PCA) was attempted in the hope it would allow a dimensionality reduction, however it failed

to identify any components that contributed to more than 30% of the variance. We therefore abandoned PCA. We also carried out simple scatterplots comparing *AlbumPopularity* against all of the numerical Spotify variables. Unfortunately not much correlation was shown in these plots. However the useful information we did take was that *AlbumValence*, *AlbumSpeechiness* and *AlbumAcousticness* had a clear influence on *AlbumPopularity*.

3 Task 1

3.1 Variable choices

The task initially required consideration of which variables to predict with, and what to predict. As mentioned in the EDA, *AlbumPopularity* was selected as the variable to be predicted and artist related variables were avoided because the style and popularity of an artist can vary greatly across albums, and the albums that made an artist popular may not be included in our dataset.

The quality of the variables as predictors was examined in our EDA and through experimentation with the models, however the usefulness and validity of the predictor to an investor in the real world also should be considered. For example *ArtistPopularity* correlated well with album popularity, but is not as useful to an investor because the price of an investment in a popular artist would likely be higher, and thus the expected return on investment lower.

Release month also proved problematic. Our analysis suggested that being recorded as having been released in January indicated a significant reduction in expected popularity, however this was likely due to the decision to record unknown release months as January. As a result this information is useless to an investor, since they would be able to select which month to release in and it would therefore be known.

Genre was more useful to an investor, however it needs to be considered with the caveat that it may only indicate a popular style of music rather than actually increasing the music's popularity. Simply put, declaring a folk song to be rap probably would not boost sales as much as actually making a rap song.

3.2 Model choices

Several styles of model were considered. These included K Nearest Neighbour (KNN) regression, ridge regression, linear models and generalised linear models. Initial work revealed that ridge regression models and generalised linear models were not suited to our task. Experiments with ridge regression suggested a model extremely close to the null model as optimal. This was not considered further as it essentially did not provide useful information as to what kind of music would be popular. Generalised linear models were abandoned as they required us to heavily edit our data to meet the strict requirements for a log link, and we decided that this would not be an effective use of our time. The remaining two styles of model were KNN regression and linear models. These were our primary focus as initial research had indicated relatively strong performance.

3.2.1 Linear Models

We initially performed a best subsets regression (BSR), comparing *AlbumPopularity* to sets of up to 9 variables chosen from the dataset. The BSR enabled us to identify which were the best performing variables with *AlbumPopularity*, and therefore gave us a focus for our linear models. The quality of the model was assessed primarily by examining the sum of square error across all 12 test albums. This gave us a ballpark estimate of the quality of the model, so extremely ineffective models could be ignored and better performing models were then analysed in more depth. The 9 variable output from the BSR was also the best model we attempted, outperforming even models that took into account *ArtistPopularity*.

Our best performing model had a Total Square Error of 1047 over the provided test values, and a mean absolute error of approximately 8.25. This was the best of any model we tried, outperforming all the other linear models and all the KNN models we experimented with. The predictions made for the test dataset are

below, rounded to the nearest integer. The worst prediction was off by 19, a very significant error. However most were within 9 points of the real value. This model made use of the variables *AlbumAcousticness*, *AlbumValence*, *AlbumDanceability*, *isPunk*, *isBlues*, *isJazz*, *isRock* and *isPop*. *IsGenre* variables refer to the aforementioned indicator variables for the presence of super-genres.

Album Name	Real Popularity	Estimated Popularity	Difference
Absolutely	48	57	9
Blue Lines	61	66	5
Dig Your Own Hole	49	61	12
Disc-Overy	50	56	6
In Dreams	54	47	-7
It Takes A Nation Of Millions To Hold Us Back	54	63	9
London Calling	71	52	-19
Meddle	54	64	10
Original Pirate Material	58	58	0
Shirley Bassey	23	31	8
What Went Down	67	61	-6
Yoshimi Battles The Pink Robots	63	56	-7

A model that only made use of mean album information (the musical data) was tested, however this model performed very poorly, with a Total Square Error of 2224. Further examination of this model was halted due to how poorly it performed. The failure of this model indicates that genre contains important information about what makes music popular. It may help determine popularity, for example people may decide not to listen to Jazz because they assume they do not like it. However it may also simply contain information about the music not picked up in mean album data or lost in the aggregation process.

We also examined the performance of a model making use of all quantitative data available to us.The model struggled with collinearity, which would negatively impact the quality of the model when used outside of the training data, as well as limiting our ability to examine predictors on their own. In addition to this issue, the model also did not outperform the BSR model on the test dataset.

3.2.2 KNN Regression

A KNN model calculates a distance value between each input data point. It then calculates a prediction for a given song by identifying the k nearest neighbours of the song and returning the distance weighted average. The distance function we chose for this model was L2, which calculates the distance as $\sqrt{\sum_{x \in X} x_i^2}$. Popularity was weighted by distance, so closer responses had more of an impact on the prediction. The weighting function was $\frac{\sum_{j=1}^{k} a_j^{-1} y^{i_j}}{\sum_{j=1}^{k} a_j^{-1}}$. Data was also standardised to the same scale to ensure that no element of the model was overweighted. The model was tested with several sets of variables. The approach that removed all popularity data, *AlbumAvgDuration*, *AlbumKey*, *AlbumSpeechiness* and *AlbumTimeSignature* had the lowest total square error, of 1106 at k = 2. This was a similar performance to the best linear model. Other KNN models were tested, however they all performed worse than the aforementioned model.

A KNN approach did not yield a significant improvement on linear models. Furthermore, a KNN model gives very little insight into what makes music popular, instead giving a means to estimate if a given song will be popular. The high dimension of the dataset also caused issues, since it ensured a high distance between each input data point, undermining the method.

4 Task 2

4.1 Introduction

As stated before, Task 2 centred on the recommendation of new music to listeners. We chose to generate a system based upon the last song an individual was listening to, rather than a series of songs. We did so as we only had access to a finite database of songs and would have had to fabricate a 'listener history' from the dataset, which may bias our results. Therefore we focused on creating a system which recommended a list of five tracks tailored specifically to the last song listened to. As studies have shown, listeners prefer music that sounds familiar to them[2]. We did however include an element of randomness in our process to prevent complete repetition of song type.

4.2 Selection of Variables

An obvious start would be to base our recommendations on the popularity measurements provided in the database. However, we did not want our system to be biased towards the tastes of the whole database, we would rather it reflect the true tastes of the listener. Hence we only took into account popularity when reducing down our final list of recommendations.

Some of the variables from the Spotify data were categorical and would require a key to be meaningful. We opted not to incorporate them into our model because it would disrupt results in an unpredictable way. The variables representing mode, key and time signature were omitted as they were found to be irrelevant in our EDA. Mode for example, was a binary variable denoting whether a track included more notes in minor or major key. Upon analysis this, as well as the other two variables, bore no evidence to suggest that they were a significant factor in listener preference.

We therefore found that the most relevant variables relating to track features were *TrackValence*, *TrackDanceability*, *TrackAcousticness*, *TrackLiveness*, *TrackTempo*, *TrackLoudness*, *TrackSpeechiness* and *TrackInstrumentalness*. The analysis of the correlations revealed that none of the variables are strongly correlated. Hence, we decided not to omit any of the selected variables as we could not afford to lose any information the dataset provided us with. This was because of the small size of the dataset, in terms of variables, as well as records. Genres were also used to provide more tailored results to the listeners.

4.3 Subsetting the Data

Before running the algorithms it was decided to subset the data. We did not include songs from the same album. This prevented a cyclic recommendation system in case multiple iterations were ever performed. We also considered the fact that a listener may want to hear music from more than one artist. In doing so we ensured that songs from different albums were still familiar to the current song. Note that it is possible for someone to get the same artist from a different album.

The next step was to make use of the created genre indicator described in 1.2.2. If the input track belonged to one of the 8 albums that did not fall into our super-genre categories then genre was not considered for song recommendations. However for the vast majority that did, we narrowed the recommendation-set by deleting all tracks which did not have a genre in common with the input track. This step allowed us to minimise the artificial similarity of the tracks which was based only on the limited feature measures provided in the dataset.

We considered time and concluded that the release date of the song is important for listeners and that the model should consider release date. For instance, a rock song from 1970 sounds very different to one from 2010 even if the genre and some of the features are similar. Our EDA also showed that the features of popular tracks vary through time. However, we were careful not to lock the user in a time specific bubble. This was done by subsetting our recommendation-set into two distinct subsets: one containing songs released within 10 years of the input and one with songs outside this range.

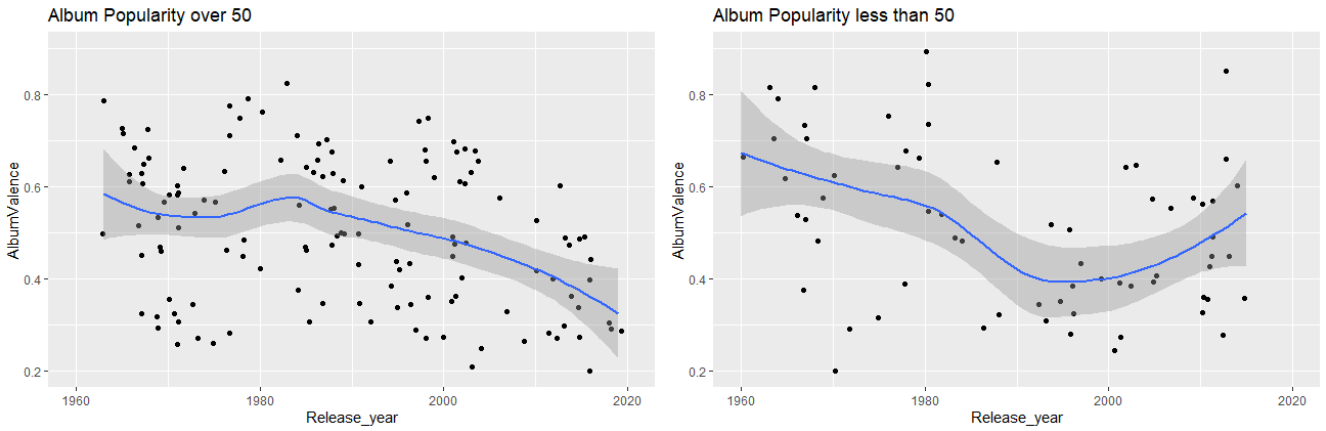


Figure 1: Popular songs have decreased in valance over time, unpopular songs have not shown this trend

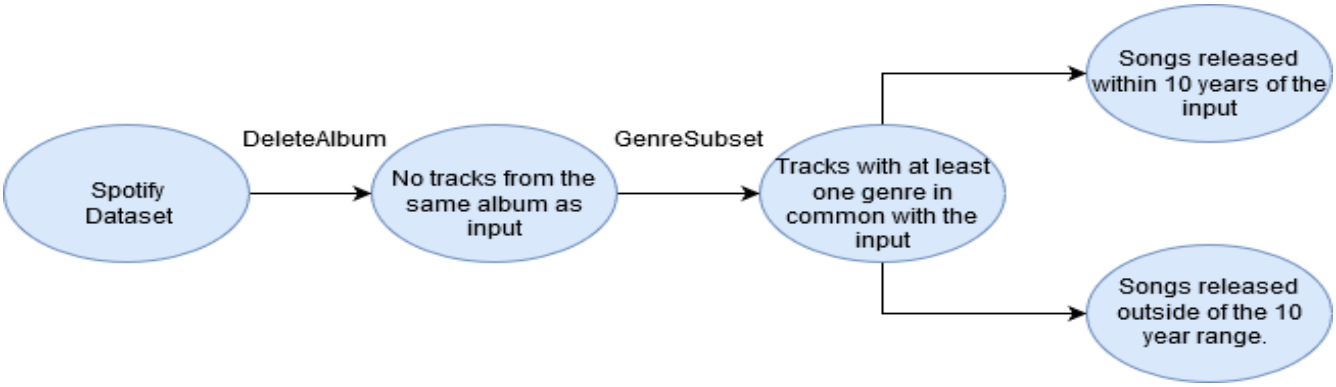


Figure 2: Subsetting Flow Chart

4.3.1 K-Nearest Neighbours Classification Algorithm

Now we have defined a recommendation-set we chose to run a KNN classification algorithm to produce our recommendation list. The KNN classification worked in a similar manner to the KNN regression model described in Task 1, however it simply returns the nearest neighbours, rather than averaging their responses. In this model, we set k = 5. Performing KNN classification with the selected continuous variables listed above in 4.2, we chose 10 songs in total, 5 from each subset. This approach meant that the recommendations can differ for the same track. We used the standardised continuous variables (a.k.a the track-feature variables) as the closest neighbours in category variables may be meaningless. The standardisation was performed before subsetting to reduce the effects of alternate scaling methods. Our approach ensures that the songs provided by the system are similar enough to sound familiar and that some, but not all, of them come from the same music era as the input-track. The KNN algorithm therefore returned 10 songs, half of them released within 10 years of the original song and half released outside of this range. An array of 5 songs was then sampled from the KNN output with the probability of being chosen dependent on the *AlbumPopularity* such that $P = \frac{AlbumPopularity}{TotalAlbumPopularity}$.

4.4 The Clustering Approach

A different approach we considered was to cluster the data and form a recommendation-set based on tracks found in the respective cluster. Clustering is a popular approach in machine learning however we didn't want to restrict the recommendation-set to a strict selection of tracks. Performing the clustering on the subsets used in our system would create a very artificial division of the songs. Hence, we concluded

that in this specific case our system of subsetting the data would result in better recommendations as it takes into account both the continuous and categorical data.

5 Limitations

5.1 Task 1

There were a number of significant limitations and issues with the dataset. One major issue with Task 1 was that popularity was measured per artist, or per album, rather than for individual tracks. Track data therefore had to be aggregated over albums. This caused problems because albums often have single hit tracks that are much more popular than others, and the features that made that track popular may be diluted by aggregation. This also significantly shrunk the dataset, as there were only 211 albums. The reduced dataset may have caused issues with outliers and other anomalies.

The reduced dataset also caused issues with genre specific analysis, as there were not many albums in certain individual genres. This was compounded by the extremely large number of specific genres included in the dataset, which further fractured it. Grouping specific genres into super-genres helped to mitigate this issue, but could not entirely solve it. Certain super-genres still had too few associated albums, which may have caused further issues with bias and random error. Furthermore the approach used to group them, necessitated by the number and format of specific genres, may have missed styles that should have been aggregated into certain super-genres.

5.2 Task 2

Genre aggregation was not a significant problem for Task 2, however the issue with the size of the dataset carried over. For some of the albums, the algorithm was unable to find any tracks with both the same genre and a similar release date. In this case, we decided to use all the songs in our recommendation-set. As a result of this the listener may get recommended tracks which are less similar to his choice but the tracks should still follow a similar feature pattern to that which is present in the provided track. When the set of such similar tracks was too small and the KNN algorithm could not find the 5 most similar songs we decided to narrow the final list from which we sample recommended songs. We did so to maintain the pattern of recommendation and allow for some measure of familiarity.

The process of standardisation is best suited for variables with Gaussian distribution. In our dataset, the distribution of some variables such as *TrackSpeechiness*, *TrackAcousticness*, *TrackInstrumentalness* and *TrackValence* are clearly not Gaussian (the plots of each variable can be found in the technical appendix). Despite this, we decided to standardise continuous variables to avoid a situation in which some variables have too much influence over the recommendations because of their distribution and range. (Standardisation was also used for Task 1, with similar issues).

6 Improvements

6.1 Task 1

The obvious way to improve the dataset for Task 1 would be to provide popularity data about individual tracks on the album. This would allow us to focus on which tracks in the albums makes the album itself popular, and therefore from this we can get a more focused idea on what type of music is popular, as we would not have to aggregate over the album.

As we are looking to recommend artists for the company to invest in, we must consider that if an artist has a high popularity it will cost more for the company to invest in them. Therefore an improvement would be to look at the difference between artist and album popularity, and see how this corresponds to the cost of investment.

6.2 Task 2

In order to improve the outcome of the system in the future, we would recommend providing a much larger database of tracks. The KNN algorithm had to be run on a subset of the dataset to provide desirable results, which narrowed the learning-set for the algorithm. More data would allow more effective recommendations.

It would also be useful to provide a longer list of the tracks that the user had previously listened to. This would require the algorithm to be more complicated to implement, in order to accurately measure the taste of the listener and carefully deal with any outliers. However, having this data would result in much more personalised recommendations with more controlled variety and unpredictability. Moreover, additional variables such as "length listened to track" would be helpful, as an indicator of how much the user likes the recommended tracks.

7 Conclusions

7.1 Task 1

The original goal of the task was to identify which albums would be popular in order to assist the client's investment strategy. Popularity should therefore be considered insofar as it relates to expected number of listeners. Our analysis suggested that the relationship between the predicted variable and the number of listeners was exponential, which should be considered when using the model. Our model implies that the more danceable a song, the more likely it is to be popular. An album being the genre of Blues or Rock also contributes positively to popularity, whereas following variables contributed negatively: valence, acousticness, Punk, Jazz and Pop. However these values are not independent, for example valence contributes negatively to popularity, however it also correlates positively with danceability. As such this model is best considered as a whole. The price to sign an artist is likely affected by their popularity, this should also be considered when investing, though it is not used to predict the popularity of a given album. The coefficients of the linear model are below:

Variable	Danceability	Valance	Acousticness	isPunk	isJazz	isPop	isRock	isBlues
Coefficient	40	-26	-15	-10	-14	-7	9	7

7.2 Task 2

The final outcome of Task 2 is an output list of 5 tracks. These were sampled from the list of 10 tracks selected by the KNN, with probability based on *AlbumPopularity*. The task given to us was very subjective and its performance could not be measured objectively. We focused on recommending songs which would be unknown to the user but sound familiar at the same time. It was not possible to test the system on real users, so the system could only be tested by looking at the recommended tracks and comparing them with the recommendations on Spotify. Given the variables we obtained from the dataset, we did not want to recommend songs that were too different, as we had little knowledge on the listener's overall taste.

References

[1] Spotify: Spotify API *https://developer.spotify.com/documentation/web-api/reference/albums/get-album/*

[2] Carlos Silva Pereira, João Teixeira, Patrícia Figueiredo, João Xavier, São Luís Castro and Elvira Brattico. Jay Pillai, Editor: Music and Emotions in the Brain: Familiarity Matters, *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3217963/*