How to Have a Number One the Easy Way

An analysis of the music industry using data mining techniques

University of Chicago
Spring 22' MSCA 31008 Data Mining
Authors: Ben Ossyra, Dominique McBride, Peter Fuentes Rosa, Peter Pezon

Introduction

Music has become one of the most expressive forms of art for today's society. The music industry involves the production, distribution, and sale of music in a variety of forms with a particular focus on streaming platforms such as Apple Music, Pandora, and Spotify to name a few. Streaming platforms has made it easier to independently distribute and consume music from a variety of artists. It has been estimated that 137 million songs are produced every year. More specifically, 22 million of those tracks are ingested into the Spotify streaming platform of which 1.2 million are released through major labels.

Spotify estimates that more than 60,000 new tracks are uploaded to their platform every single day[i]. In response to large volumes of streaming, record labels have signed more artists than ever before. In 2017, it was estimated that major record labels spent over $4 billion on A&R alone and signed close to an average of 2 new artists a day[ii]. It is evident that the days of discovering new artists via demo tapes are over and companies need more sophisticated methods to identify and sort through high volumes of music to focus on songs that will yield the greatest success.

Business Case

A small independent record label has hired a team of data scientists to help them identify the next big hit. With limited resources to invest in A&R, the company has tasked the team to develop a model that could be used to reduce the time it takes to review music and determine what aspects of a song are essential for producing a chart-topping hit. Where should the investment be made?

The initial question that comes to mind is quite simple, what makes a hit song? Is there a certain combination of audio features that is similar amongst most hit tracks, or is the star power of the artist and the marketing dollars the most predictor of success? The cynics in the industry may lean towards the latter, but any regular listener of the radio can probably identify certain features of a big hit.

Our goal is to use data mining techniques to develop a model that will help identify potential hit tracks. Using supervised learning methods, we are hoping to narrow the pool of artists down to employ more traditional evaluation approaches. We will look at both the audio features of songs (danceability, energy, loudness, etc.) as well the features of the artists (number of releases, chart months, award nominations) to determine if there is a magic formula to identify a hit song.

Dataset

For this project, we combined data from the following sources to produce our final dataset. The contributing sources include the following:

1. Spotify Charts dataset Kaggle[iii]
2. Charts 2000[iv]
3. Spotify API
4. Spotify Daily Charts
5. MusicBrainz
6. Alcrowd

The original data source was collected from Kaggle and built on from additional sources to create a robust workbook that could be mapped out as needed for the next phase of the project. Additional attributes were added by extracting data directly from Spotify utilizing an API to append track and artist-

level information. In addition, we appended historical track charting data from MusicBrainz with the intent of evaluating charted tracks during the past 5 years. Finally, we added 15,000 non-charting tracks to the dataset from Alcrowd and Spotify Charts to provide more analytical depth, and de-duplicated the final dataset.

Spotify is a database and source of truth. It is the largest streaming service, with a catalog of 82 million tracks. We also rely on MusicBrainz, a crowd-sourced and open-source database. We rely on Spotify for audio fingerprints and artist catalog data. We also rely on the Spotify Daily Top 200 charts for random sampling songs, and their charts are partitioned by country and global charts.

For charting, we rely on Chart2000, which produces a global top 50 chart by aggregating hundreds of charts worldwide and normalizing the track positions the respective country's total music industry revenue using data from the International Federation of the Phonographic Industry, which collects the total revenue of the music industry by country. You can read more about the Chart2000 methodology on their website[1].

Our dataset was constructed by collecting tracks from Chart2000, Spotify Daily Charts and MusicBrainz' GRAMMY lists. There is a bias, because there isn't a true random sample of songs, and the sample set is heavily based on the Global Top 50 and the Spotify Daily Charts. That having been said, music content has a "long tail," and only the same hundred or so songs are played ever.

There is bias by relying only on Spotify Top Daily charts, because there are other services where music is streamed and shared, such as YouTube, SongCloud, Deezer, Tiktok, etc. Our assumption is that, when relying on the Spotify daily charts, although the consumption rate for Spotify may be low or biased in a particular country, it may still reflect the general consumption in that country. This is an assumption that should be validated.

It should also be noted that the U.S. market accounts for $8B or 34.6% of the recording industry total revenue of 23.1B, by far the largest market.

Our focus was creating a balanced dataset relevant to our analysis — determining songs that will chart. We needed a data source that of charted songs. It's difficult to ingest hundreds of sources of data, so we found an aggregator of global charts. Now we have a dataset of charting songs, and we needed to include songs that have not charted.

To construct a set of track IDs, we collected songs from Chart2000 and used Spotify Search API to retrieve track IDs, collected songs from MusicBrainz's GRAMMY release groups, and matched those with Spotify with the Search API. To include songs that have not charted, we sampled randomly from Spotify's Daily Charts to ensure a class balance of 20%. Although this chart is biased, it does serve a purpose. Generally, music consumption has a very long tail, as most songs have under a hundred plays. We wanted to include songs that were somewhat popular. After compiling a list of track IDs, we used Spotify's bulk APIs to retrieve metadata and audio features.

In order to merge data sources, we used fuzzy matching to match artist names and track names. When we retrieved a list of songs from our data sources, the source datasets did not contain a Spotify Track ID. We needed to search Spotify for the track. We didn't trust the first results returned by Spotify (for example, Chart2000 lists "The Backstreet Boys," but on Spotify the band is named "Backstreet Boys," and the first results for "The Backstreet Boys" were children's lullabies), so we needed to match the search results returned by Spotify Search API to make sure we were selecting the right result. We

[1] Chart2000.com: Music Charts 2000 – 2021 Data Collection Methodology. https://chart2000.com/about.htm

matched with fuzziness to ensure for variations in changes to the track name or artist name across data sources (e.g., T-Pain and T. Pain).

We also created additional data sources to create new features that we thought may be relevant for analysis, such as features that could represent an artist profile, such as the number of releases by the artist. There are a number of data sources we could include to create new features, such as song samples, or deeper charts, genre charts. Ultimately, resourcing was the limiting factor in what data sources were collected.

The dataset is composed of music from all genres, and contains genre features, but modeling on genre features did not perform well, and creating models based on genre features did not make sense because genres are not mutually exclusive; genres overlap. It may not be reasonable to train a model on non-mutually exclusive variables. Modeling genre features may introduce a bias to predict a class as charting or not charting based on genre, e.g., classifying pop genre songs to be more likely to chart than other songs; or trap songs, which have so far charted less often. Genre popularity trend and that trend is not being captured in this model. It would be better to generalize on patterns found within track's audio features that transcend the definition and contemporary popularity of genres. A better solution may be to train models on genres: for each genre, training a model on a subset of tracks from a genre.

Our final dataset contains roughly 15,000 tracks (rows), and an upward of 60 features (columns). Some of the track descriptive columns (track_number, disc_number, etc.) were not relevant to our analysis and were removed. The rest of the data was sorted into 2 main categories, track description, and artist historical performance. With this came several modifications to clean the data for use, but with more details in the data preprocessing. For the purposes of the project, we will consider a track that makes it onto the Billboard charts as a commercial success. This will be our target variable.

The original Spotify dataset is compiled from recordings nominated for a Grammy, with is around 535 tracks. Also, there were tracks that have entered the Chart 2000 global aggregated monthly song chart, which is around 3,300, and a random sample of 1000 from Spotify daily charts. The dataset is de-duplicated on track id and limited to the years 2000-2021. The dataset is small, around 4,237, but balanced across the three categories. Later, additional tracks were added to reach the 15,000 mark.

The dataset is limited to 2000-2021 because of our data source, Chart2000, only tracks charts from 2000-2021. Additionally, the Music Industry also had a seismic sea change in the 2000s when the markets for CD jewels collapsed and piracy and streaming music services like Napster, Pirate Bay, Grooveshark and Spotify became dominant points of consumption. Previously, chart statistics were based on terrestrial radio play and volume of units sold by distributors (not necessarily retail to consumers). In the last two decades, the major recording labels have been losing gatekeeper status, owing to the success of independent labels and direct-to-platform self-distribution services; and terrestrial radio has seen rapid consolidation.

Therefore, the means, measures and reasons for a track to “chart” have changed, and so have national popular tastes, so it may not be appropriate to compare recordings from the 2010s to tracks from the 1950s. However, there may be some generalizable patterns among popular (and charting) recordings that have transcended the decades, such as 4/4-time signature and pop song structure, which has been prominent in American popular music since the 1950s.

Collecting the data took several weeks, due to complexity of collecting data from multiple sources, rate-limiting of data source APIs, and errors matching strings between sources. A full run of the pipeline can take 24-48 hours end-to-end due to rate-limiting.

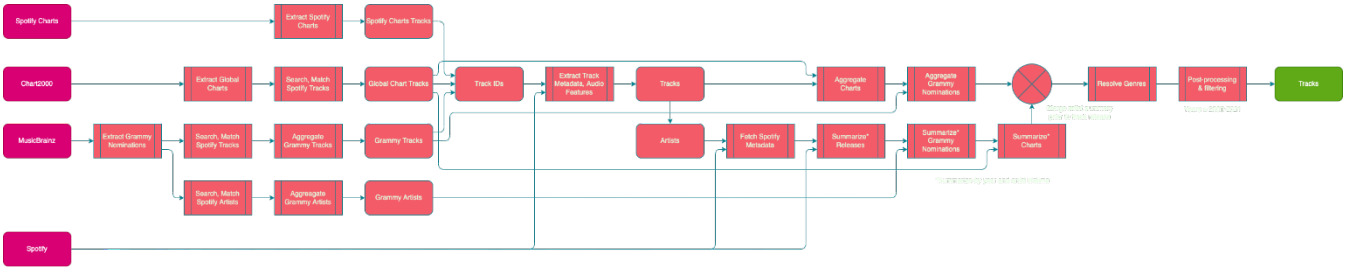

Figure 1: Data Collection pipeline

Exploratory Analysis & Data Preprocessing

To begin the preliminary EDA, we identified the key features of a song and separated them into the following categories:

Track Features:

- Audio features
- Track award features
- Track Billboard chart features

Artist Features:

- Artist award features
- Artist demographic features
- Artist chart features
- Artist releases features
- Artist genre/record label

The track features provided a descriptive view for each individual track based on audio metrics like loudness, energy, and tempo to name a few. In addition, the data also shows a track performance with the music industry's awards (GRAMMY's) and the associated charting history. These provided insights into how a song performed in the industry. The artist features also had similar functionality for our business case. Artist data like awards, demographic, charting history, genre, and record label will provide insights on how an artist performed historically and the likelihood of future songs charting.

Understanding the data

Our initial analysis of the audio features shows that in most cases there is not an obvious difference in the features of the songs that have charted and those that have not charted. The box plots show slight differences in the averages between charted and non-charted for the audio features energy and acousticness, shown below in Fig. 2.

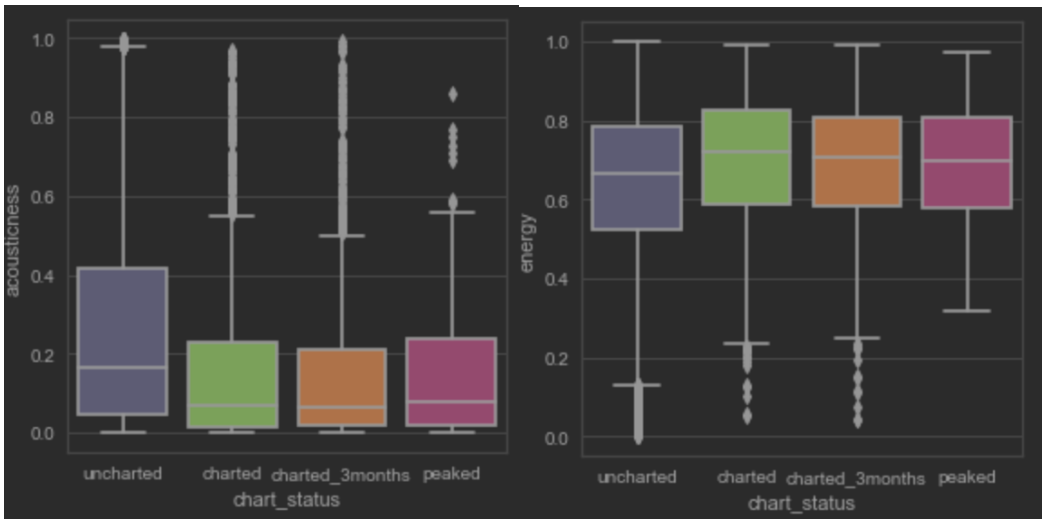

Figure 2: Box plots for acousticness and energy and chart status

Parallel Coordinates plot in Fig. 3 show there is little variation among the audio features; there is only variation in tempo, loudness, and key. There is no obvious grouping of charting and non-charting tracks on these features.

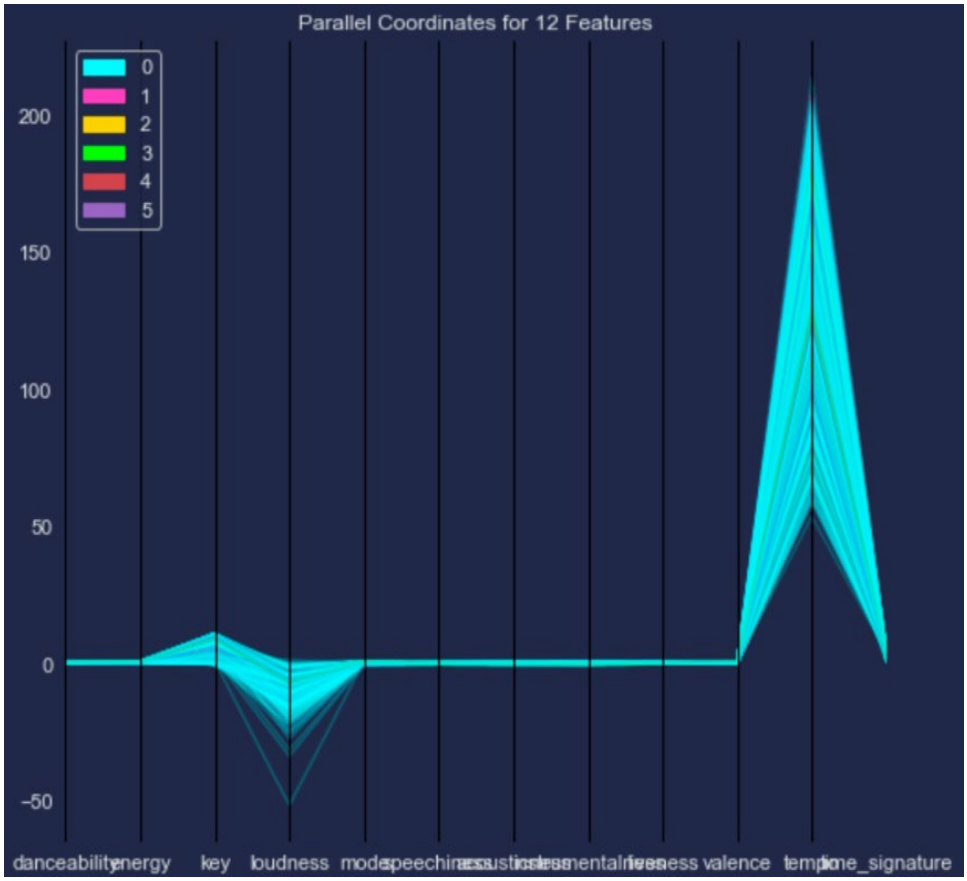

Figure 3: Parallel Coordinates of audio features

The distribution plots in Figs. 4 and 5 indicate that danceability, energy and valence are normally (fairly) distributed. Tempo has 3 distinct peaks. Loudness and speechiness are both very tightly distributed.

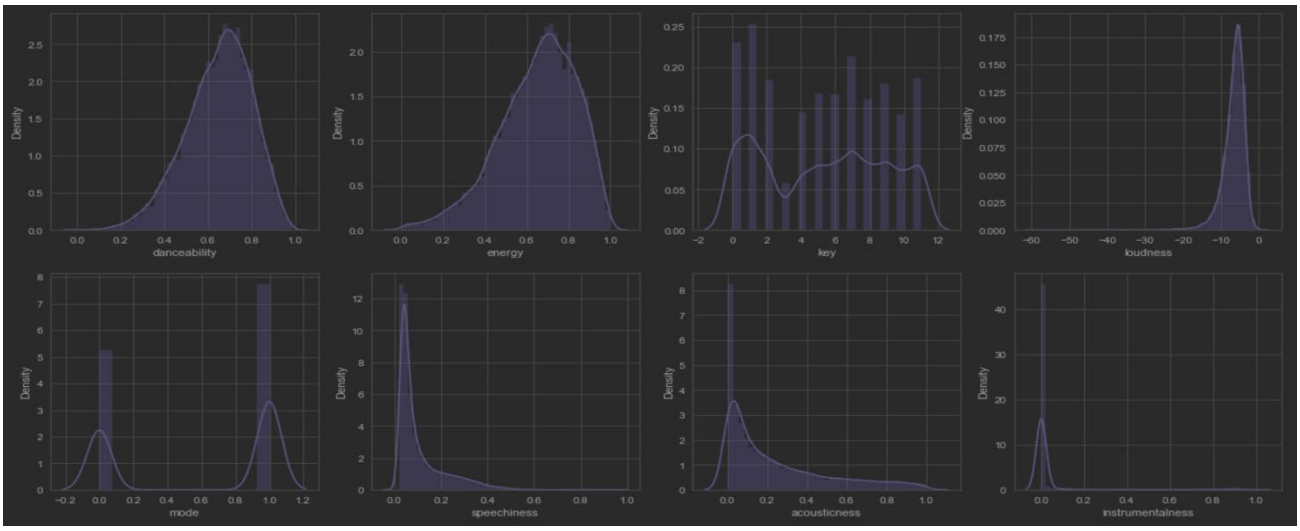

Figure 4: Distribution plots for danceability, energy, key, loudness,

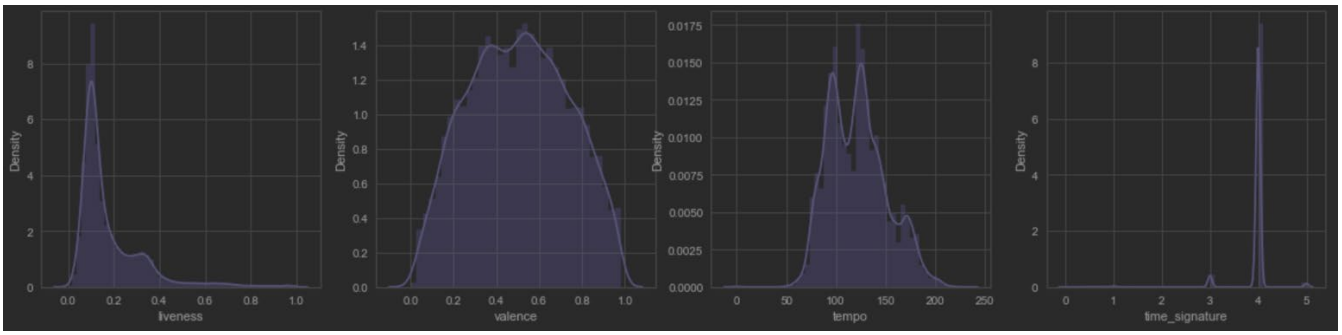

Figure 5: Distribution plots for liveness, valence, tempo and time signature

As expected, the box plot for the artist features in Fig. 6 shows more distinct differences between the charted and uncharted songs. Artists that have previously charted tracks are much more likely to chart again and chart for a longer period. Additionally, artist wins seem to be significantly distributed independently from uncharted to charted and charted in the past 3 months.

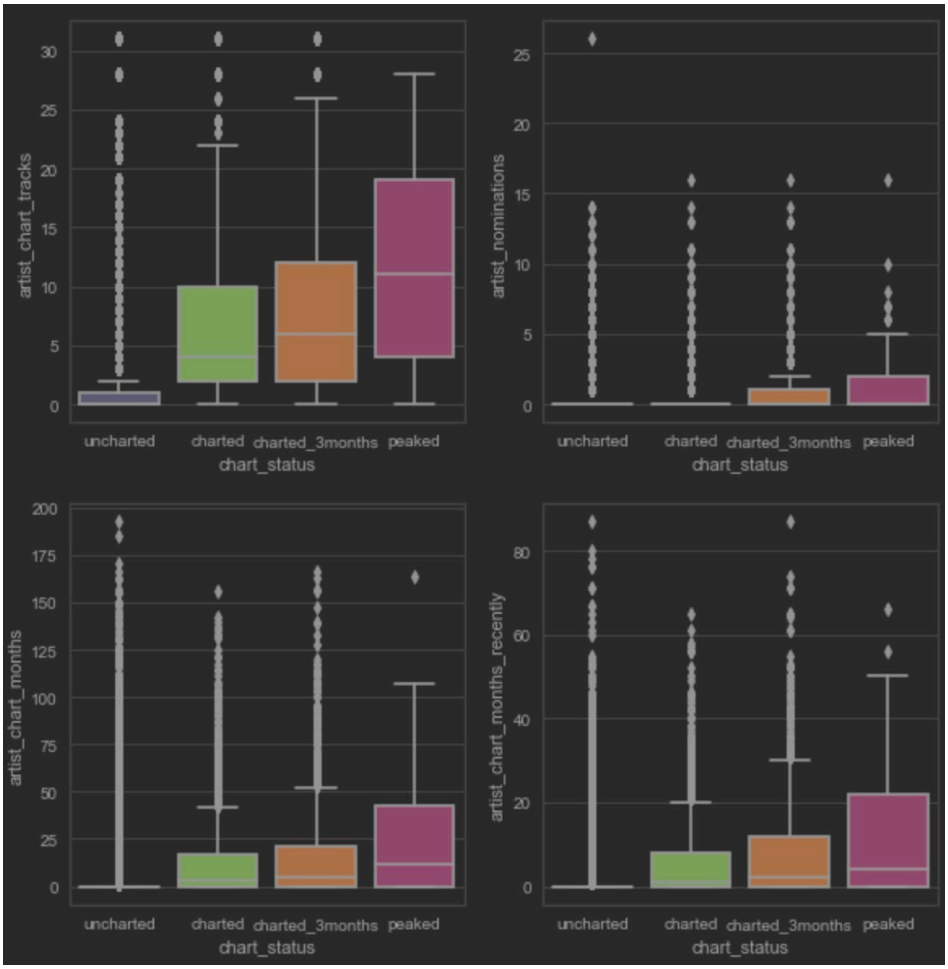

Figure 6: Box plots for artist features artist chart tracks, nominations, chart months and chart months recently

Parallel Coordinate and RadViz plots in Fig. 7 show a lot more variance among the features, particularly number of releases and chart months.

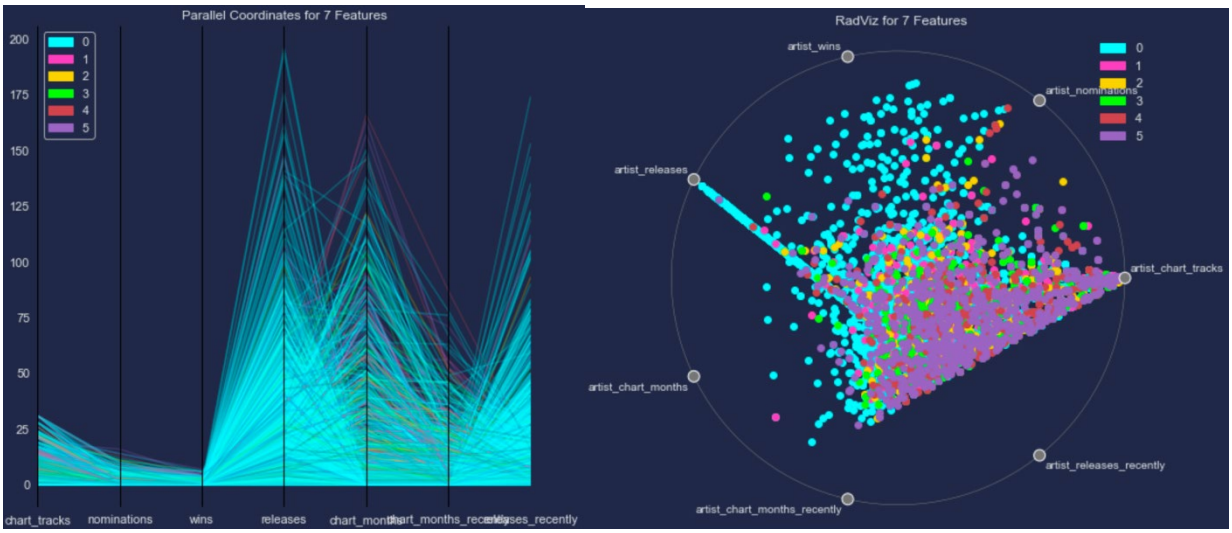

Figure 7: Parallel Coordinates and RadViz plots for artist features

We constructed a Random Forest model with all audio and artist features to determine feature importance, shown in Fig. 8, and found that the most important feature was *artist_chart_tracks* and *artist_chart_months*, which measure the number of tracks that an artist has had on the charts, and the number of total months that an artist has been on the charts, respectively. This is an issue because our model should predict songs based on audio features. Our objective is to provide insight into relatively new artists, not already established and popular artists with a history of charting.

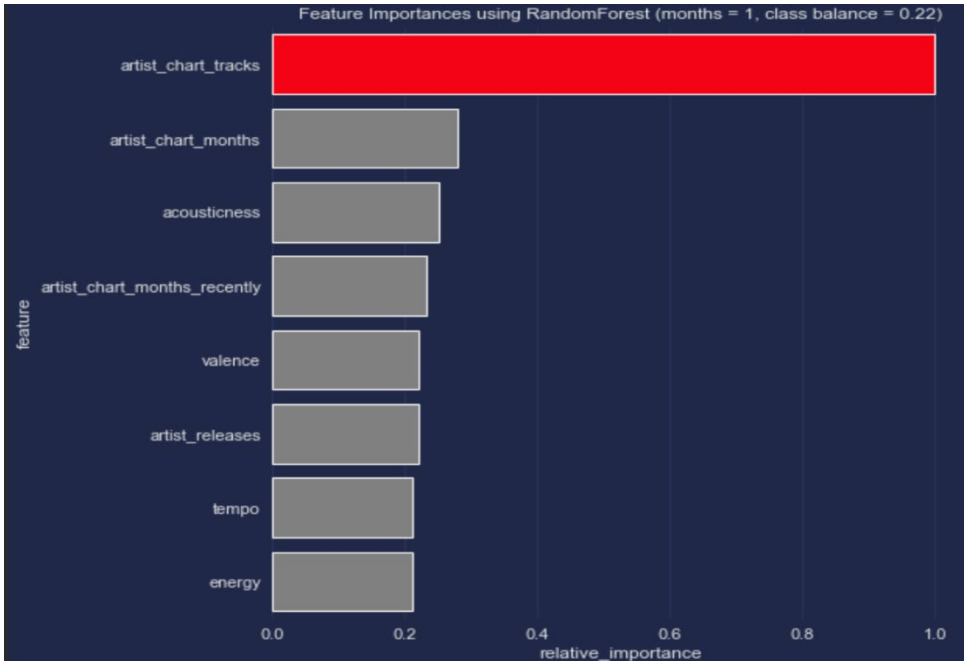

Figure 8: Relative Feature Importance based on Random Forest model

Prepping the data for analysis

Data preprocessing was one of the most challenging tasks needed to produce a functional dataset. Once the data from all the sources were collected and merged, we performed several methods of data cleaning to include:

- Data Revision: As the team began working with the data, several things became known that required multiple iterations of data clean up. This included errors in the original dataset such as false track popularity based on karaoke songs and genre distribution. The distribution was niche to each individual song and was unusable.
- Missing Data: When appending charting history data to the tracks, there were several issues in the process. Mostly, we could not get accurate charting history data for all tracks that were classified as charted because the data source did not have all the information needed. In addition, part of our dataset included tracks that do not have any charting history because the tracks themselves were classified as non-charting for the purpose of our project. To work around this, we created additional metadata for those tracks that had no charting history or were non-charting. Finally, some datapoints were missing in our overall dataset, so we used data cleaning methods like adding the mean value of the feature(column) for missing data or not using rows with missing data for model training/test purposes.
- Data Normalization: When preparing our data for each model, we noticed that the data had different scales numerically. While some features were binary, others were significantly high, like sales and revenue. For this, we normalized the dataset to ensure that the data was scaled appropriately and thus when used for the models, the results would be meaningful.

This process equated for most of the project time as the models initially yielded unfavorable results. Whenever this outcome occurred, we had to revisit the dataset to ensure it was as clean and normalized as possible. Throughout the modeling process we continued to identify data irregularities in the dataset that we addressed and scaled for optimal performance.

Feature Selection

The tracks datasets from Spotify have track level with metadata such as artist name and track name, genres, etc. Spotify provides high-level summations of audio features, such as tempo, danceability, speechiness, etc. Spotify also offers EchoNest feature analysis, which breaks down a track on a discrete level. We went with high level summations due to experience and resourcing.

From professional experience in the music industry, there was a suspicion that charting songs and GRAMMY nominations have very little to do with the quality of the songs, and more to do with politics and marketing. We tried to incorporate other features that were not related to audio features, such as charting statistics, number of releases. We attempted to collect information about the record label, such as whether the track was released by an independent or major label, suspecting that major labels have a bigger budget to push a song on the charts. (This feature was not used due to limitations in collecting this information.) The point of collecting non-audio features was to determine how much less important audio features were compared to other features and attempt to draw a conclusion about influence from marketing and politics.

Regarding the feature selection for the models, there were various analysis done in EDA that helped us determine which features to select, including recursive feature extraction, and stepwise feature elimination on p-values. We ended up training over 280 different models. We trained many models on

different feature sets, and with brute force found a model and feature set combination with the best performance, using AUC and precision as performance metrics to evaluate the models.

It was important to use audio features and avoid features that interfered with our business case. For example, we ignored features such as how many months the artist previously had a song on the charts, because our business case is to identify new and up-and-coming artists. (Creating a model with the absolute best performance was not our objective.)

We also selected features that were complete. We did not use any features in our analysis that had missing data. This was somewhat simple to control, since we compiled the dataset. However, there were data points that were difficult to collect and were not complete, such as artist gender, artist country of origin, and recording label; these features were not included in any model.

On a technical level, we used a stepwise forward and backward selection on p-value and recursive feature extraction on a Random Forest model to select features. We compiled a list of feature sets, and trained several models on different feature sets, and picked the model with the best precision score to select our features.

Analysis & Modeling

For our purpose, we provided several models that give alternative ways to address our business case. These models take different methods of analyzing our data and trying to address our questions to see which method has the best results for success. Success rate is determined by the evaluation metrics in each model to see the accuracy, precision and, F1 score to see each model's predictive power to ensure best outcome for our record label's goals of seeing what songs are going to chart.

Clustering Analysis

Before modeling, we conducted a Clustering Analysis to see the relationship between the tracks data. Clustering Analysis is an unsupervised learning technique used to segment data into a set of homogeneous clusters of records with the purpose to provide insights. There are several clustering methods, but for our project we utilized k-means, which is a record allocating method that separates the datapoints into clusters according to the distance from each other. As an initial step in our project, we conducted a Clustering Analysis to identify relationships between the track data collected. Our intended purpose of clustering is to understand what the relationship between tracks are and their genre. In this, if the results were significant, use these findings to guild are next steps of creating a subset of the data to push into modeling with the goal of creating better resulting models.

In the analysis, we took into consideration 13 attributes from the tracks data, 12 that are used in the cluster method and 1 as the attribute we tried to classify for. These include:

danceability	acousticness	mode	key	liveness	tempo
time_signature	instrumentalness	speechiness	loudness	valence	energy

The attribute we are trying to classify for is genre_values. With this attribute, we conducted 2 kmean exercises. The first was by genre_values, which where a total of 24 unique categories, so we divided the data into 24 similar cluster and cross validated with the genre_values to see how if there was a connection between song attributes and genre. After this, we conducted another kmeans analysis to reduce cluster to see what the best amount of higher concentrated cluster groups would be to see more defined distinctions between clusters.

Classification Analysis

a. Classification Techniques Models
    i. Logistic regression
        1. The Logistic Regression model was used to provide insights into which features would be ideal for being able to predict a hit song. For feature analysis we looked at the tracks that charted for 1 or more months (label) and found that the optimal number of features to evaluate was 10 features ('danceability', 'energy', 'key', 'loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo'). A key consideration for this model was to balance class weights to account for imbalance and ensure standard scaling and set the discriminant threshold to 0.7.
    ii. Decision Trees
        1. Decision Trees are a method of supervised learning that uses the fields imported to categorize or make predictions base on how the data splits between layers. For our project, we created a decision tree model, using the same 12 features as the clustering, but now with an additional feature to measure if tracks have charted or not. This model was to see how well the model trains and classifies with the testing data tracks that have charted vs not charted.
    iii. Random Forest
        1. In the execution of the Random Forest algorithm, the team wanted to evaluate the impact of randomly selecting one subset of features over another. For example, do artist features establish better prediction criteria than audio features, should the top features of GRAMMY awarded songs be a standalone predictor or is there an ideal combination of them all?
    iv. Neural Net
        1. In the Neural Network model, we utilized the tracks data to find any underlying relationship between the attributes to see how well we can used these to predict song success. This method maps out the relationships like the connections in the brain to understand how they relate to each other. This method takes in the data by the input layer processes it between the hidden layers by learning from the data itself, with weights and bias, and exports out the output layer.
    v. SVM
        1. In the Support Vector Machines, we take our data and project it is using a mapping function that normally transforms non-linear dataset and makes them linearly separable. With our data, we applied this model method understand how the model separates our data points and scaling them and takes in new datapoints. This then groups them based on the data separation to see if it can identify from the tracks coming in, where to add the new tracks group with to see how well it can predict

the tracks coming in. In our data, the separation would be between tracks that charted and that did not chart.

b. Model Tuning
    i. In the initial phases of modeling, we experienced low precision and accuracy scores resulting in the need to fine tune our models to increase performance. We utilized Grid Search to identify the most optimal hyperparameters, Random Search to generate a random selection of values within a given hyperparameter space in addition to scaling the dataset midcycle to reduce redundancy (removing karaoke tracks that yielded inflated results). Not to mention, with low scores, we oversampled models to help improve results with additional observations in the training sets.

c. Evaluation metrics
    i. Accuracy
        1. Accuracy is one of the most direct metrics when looking at classification or predictive models. This metric stems from the confusion matrix which analyzes how well model performance is based on comparison of actual values and predictive values. Accuracy of the model is determined by summing up the true positives and true negatives and dividing those over all the predictions. For our business case, the accuracy is how well the model can classify or predict correctly tracks that are not going to chart and tracks that are not going to chart over all predictions. This metrics is important to have to see how well your overall model is performing and if it has area for improvement.
            a. The other metrics that are not taken into consideration in the accuracy score are the falls positives and false negatives, also known as Type I error and Type II error. In our business case this translates to 2 opportunities, in Type I error, we pursue an artist, but they do not chart. The cost here is time on investment and the goal should be to minimize Type I error to reduce wasted time. Type II error is that we missed an artist that made a hit song, which cost us revenue and reputation.
    ii. Precision
        1. Precision is the metric to see how well the model measures the true positives, which in our example would be artist songs that charted, and the model predicted they charted. This is the most significant metric to our business case because this will be what A&R companies will look for with songs to see which to invest in.
    iii. F1
        1. F1 score is the metric to compare models for overall performance. This takes into consideration both Precision and Recall and analyzes them to create a # representation on how well the model performs. Depending on the model complexity, this could be an effortless way to determine best model results.
    iv. AUC

1. Area under the curve(AUC) is the accuracy measurement metric that scores how well the classification was executed. AUC shows how each model scored the data in the goal to see how well it performed. We used this metric, in addition to Accuracy and Precision score to show each models overall performance.

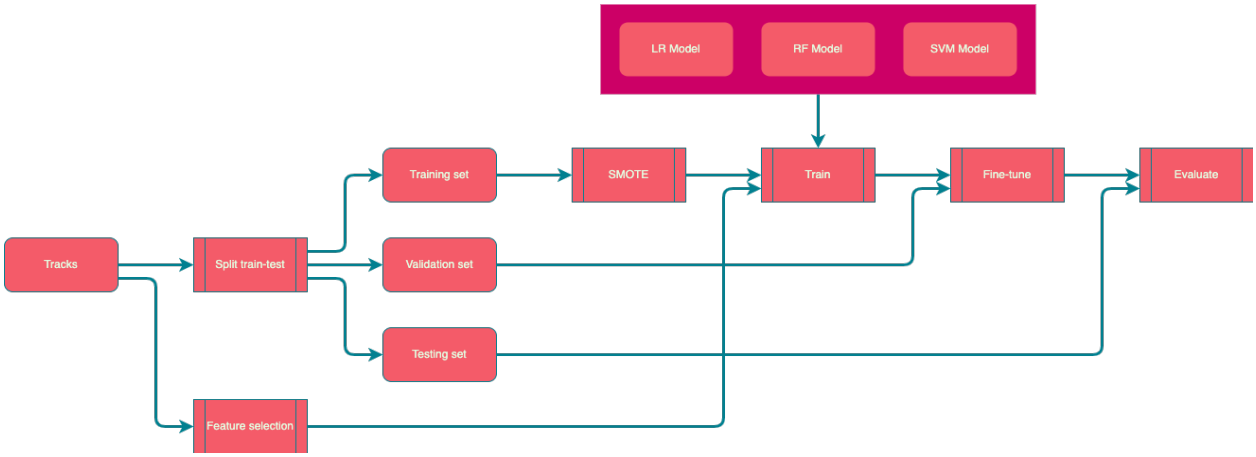

Figure 9: Classification modeling pipeline

To measure overfitting, we used a learning curve, as shown in Fig. 10 below. If our model underfit, it was indication to add more data, or reduce features. Typically, our models training and validation scores were unable to converge, indicating we needed more data or better features. We selected models on AUC and optimized precision based on business requirements – to reduce Type I error. The learning curve for a Logistic Regression model trained on audio features shows an underfitting model, indicating that the model needs more complexity or more data. Therefore, we tried carefully to add more features or use complex models.

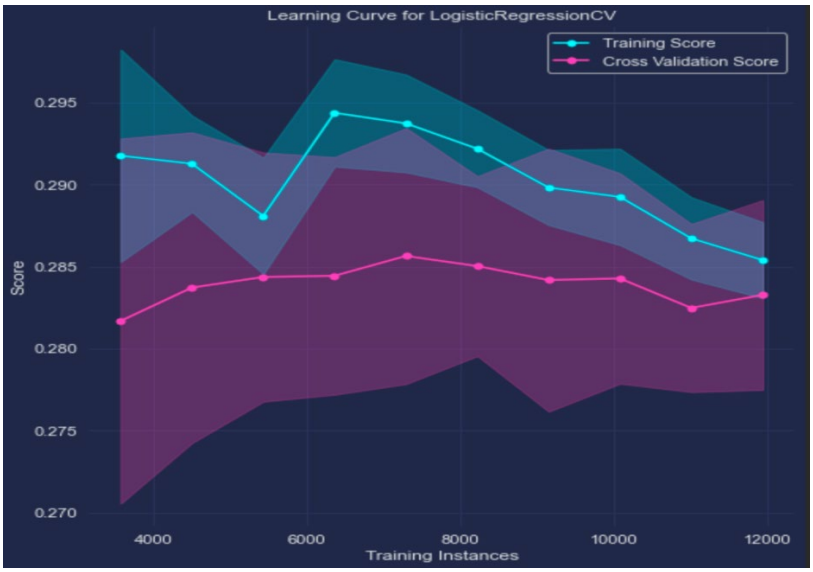

Figure 10: Learning curve shows the average training and cross-validation

We also adjusted the discriminant threshold after fine-tuning the model to achieve an optimal precision score. To do this, we plotted a chart with the average scores of a model at different levels of the discriminant threshold, as shown below in Fig. 11.

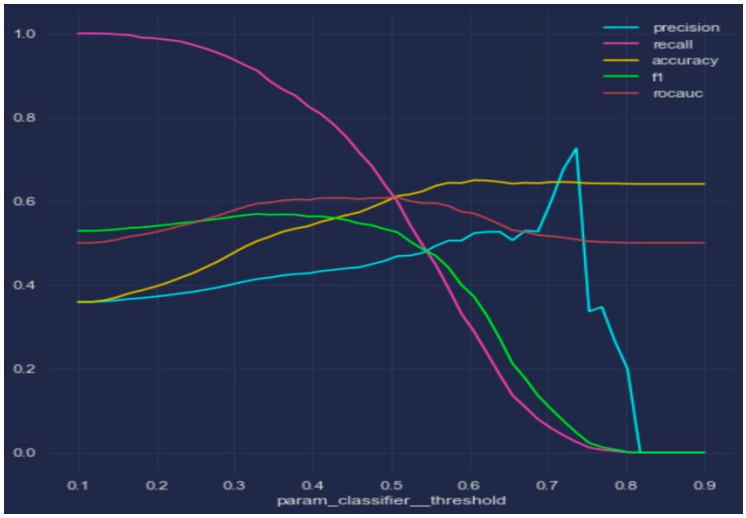

Figure 11: Performance metrics plotted against different values for classifier discriminant threshold

Results

The Clustering Analysis resulted had low metrics for the classification analysis. Based on the split between 24 genres to see if each genre had enough uniqueness to classify each track, the accuracy was only 4%, with the highest precision score for the clusters of 11%. This shows that songs attributes likeliness is not enough to identify genre and see if there is an opportunity for success to predict charts based of genres. As an attempt to see if there was any opportunity for improvement by reducing genre, we redid the clustering analysis with less genres, but the best results of cluster using kmeans was 3

clusters but because of the low curve of the elbow plot and the close to 0 silhouette graph, this reassures that clustering data by genre has no significant impact on future performance of models.

The Decision Tree Model performance correctly classified 2,977 out of 4,422 tracks, yielding an accuracy score of 67.3%, which is decent. Not to mention when looking at F1 score of 23.7%, this shows how strong the recall and precision scores. When addressing our business case, having an accuracy of 67% is not bad, but this is weighted heavily on tracks that did not chart, meaning that it classifies more songs that did not chart. Out of this, it only correctly predicted 225 out of 4,422 songs that will chart. If an A&R company utilized this model with the premise of trying to analyze 20k songs/weekly, it could find almost 900 songs/artist that could chart weekly.

The Neural Networks model initial performance was significantly low with a precision score of 21.6% and 3% F1 score. Taking this into consideration, we adjusted performance by oversampling the training data for better training the model, tested the model's thresholds and included hyper parameter tuning . This provided us with better results with an accuracy score of 60.8%, which is decent. In addition, precision when up to 26.3% and the F1 score to 34% meaning that this model cold predicts 461 tracks that are likely to chart out of 4,422, which it is roughly 10% correct prediction for track success. With our business case being prevalent, we can see how this would benefit an A&R department who could focus on just 2,000 of their 20,000 weekly tracks.

The first Logistic Regression Model we tried produced a false positive result. After re-training the model (identifying the best hyperparameters) we were able to build a reasonable model and move forward with setting a prediction of 100 out of 5975 songs (1290 charted tracks) songs would chart. Of the predicted value, 48% charted. The output is a decent result. To relation to the originating business question, this type of result would assist an A&R department's efforts to streamline their business operations. If their current operations include the review of 20k songs/week, based on this model we could present a list of 300 songs that have a 50% chance of becoming a chart toping hit. This model has the potential to make better predictions. At a minimal it will reduce the workload song reviews and potential redirect funding towards songs with greater success forecasts.

The Random Forest model identified several outcomes but with either a low precision score or unrealistic scenarios based off the dataset. One outcome was to evaluate 6 artist features ('artist_chart_tracks', 'artist_nominations', 'artist_releases', 'artist_chart_months', 'artist_chart_months_recently', 'artist_releases_recently') but produced a precision score of .05. If we were to focus on audio features alone, we could group the following 10 features together ('danceability', 'energy', 'key', 'loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo') and produce a 0.0 precision score. The model was then subject to threshold testing and hyperparameter tuning that resulted in low f1, precision and accuracy scores. Utilizing the Random Forest Classifier produced many classifications to consider but was not ideal for identifying which features to build predictions around.

After getting many poor scores in the above modeling efforts, we decided to take a different approach to segment the data by US and Global artists with fine-tuned hyper parameters and a more in-depth focus on varying SVM, Random Forest and Logistic Regression models. This refocus resulted in the training of 280 models on different variations of feature families to identify which models produced the highest level of accuracy. Models trained on US artists appeared to be the more precise than those outside the US. We were able to identify the 'Artist Profile History' is the best predictor, not audio

features, and the best fitting models included only artist features. As a result, we determined that Random Forest had the highest precision, and the SVM-RBF had the highest accuracy. See figures below:

	dataset	features	model	months	precision	recall	accuracy	f1	auc
271	US-6	Distances	SVM-RBF	6	0.105148	0.780488	0.516055	0.185328	0.638239
274	US-6	ArtistReleases+Distances	LR-L2	6	0.107229	0.723577	0.555619	0.186779	0.633226
7	Global-1	Audio+ArtistReleases	LR-L1	1	0.302418	0.712953	0.583557	0.424691	0.630473
8	Global-1	Audio+ArtistReleases	LR-L2	1	0.302418	0.712953	0.583557	0.424691	0.630473

Figure 12:Top Model Families by feature sets (Ordered by AUC)

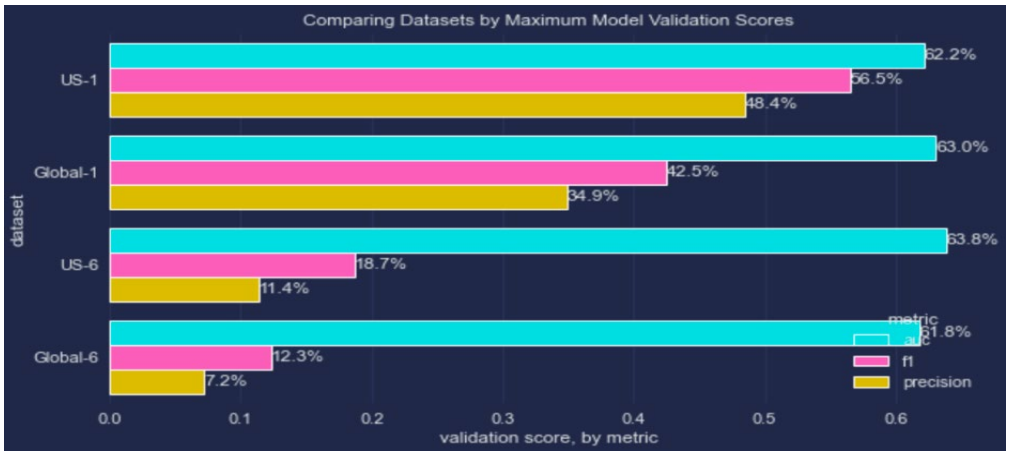

Figure 13: Models trained on US Artists (more precise)

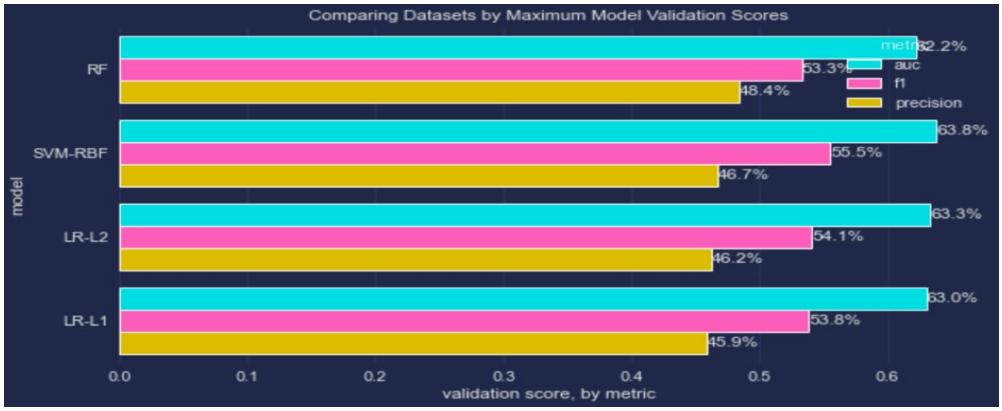

Figure 14: Random Forest models on average have the highest precision score. SVM with RBF kernel have the highest AUC score.

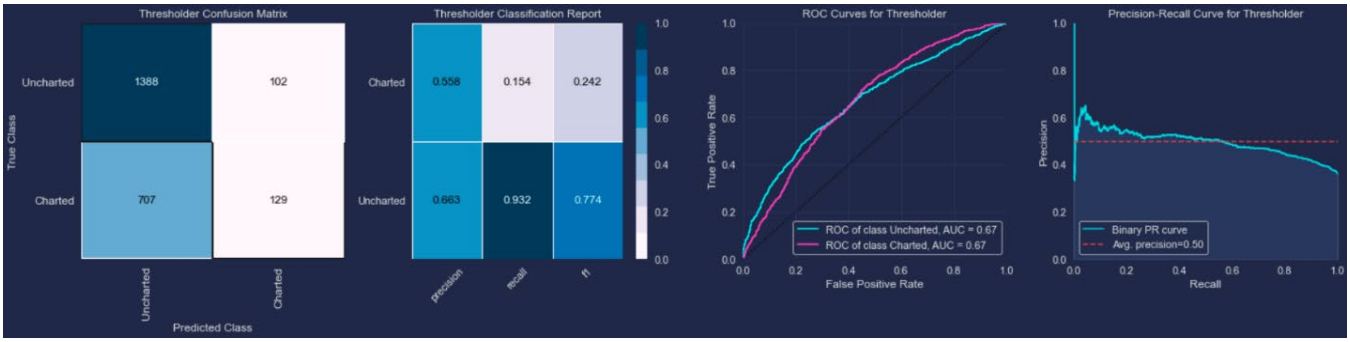

Figure 15: Fine Tuned Random Forest on Audio Features and Recent Releases

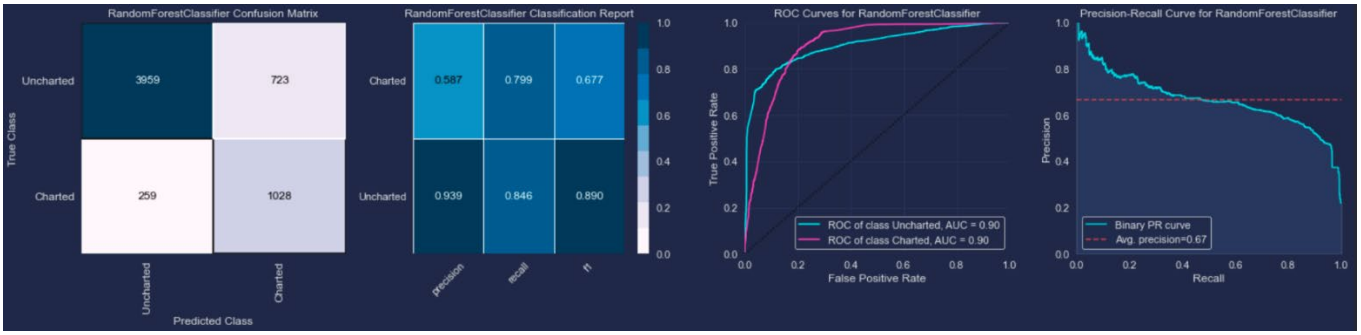

Figure 16: Artist Profile History as Best Predictor; Artist Feature Inclusion (best fit)

Discussion

As the project concludes in discovery of not so optimal results, there are some limitations identified by the team that would strengthen the models. For example, when evaluating the data set, having used Chart2000 for a global compilation of Top 50 Charts may have been a limited element. Chart 2000 may not have considered all Billboard Charts in publication such as Adult Alternative, Active Rock, and many others. Two examples of this issue are with two artists, Skrillex and Sleigh Bells, who both had songs that charted on other Billboard and UK Charts. This serves as an opportunity to expand the search criteria to more than the Top Charts and potentially using other charts.

Another issue identified during the process was with the Spotify data and how it is cataloged. During the data collection effort, we had to match artists and tracks to Spotify track ids, which didn't pan out to be a clear 1:1 relationship for many tracks due to variations in labeling. For example, Chart2000 attributed songs by the artist "Ke$sha" to "Kesha" and "Missy Elliott" to Missy 'Misdemeanor' Elliott" with poses a major data integrity issue when matching records. This issue of not being able to properly match artist to song created an instant bias that may have greatly impacted the modeling of clean data.

When looking at the results of our models and connecting back to the business, we see how an A&R department can benefit in search for the next track to succeed. Taking into consideration the precision score, we have a lower number of songs that chart and are predicted to chart, which is the main goal of these models. Using this metric, A&R will have a better opportunity of investing their resources in tracks & artist that have not charted or do not have chart history to push them forward for company success.

Understanding how the precision score is established, we know that there is a possibility of a Type 1 Error occurring. From the business perspective, in addition to looking for possible charting songs, they must watch for the company's assets by reducing investment in type 1 error from the models. This error is shown as songs are artists that are pursued because the model predicts them to chart, but do not, which result in wasted resources and funding.

In thinking about a potential next step for this project, having resolved the issues previously mentioned, the models could be enhanced to focus on certain markets instead of the industry as a whole. The models could be segmented to have a more granular concentration of a particular markets within each country. Our model wanted to focus on being able to generalize global trends but given the knowledge gained from this project and the music industry, music in today's economy has more focus on local consumption and distribution.

Additionally, for the next steps, we suggest looking for more metrics on track performance in the charts and how long it stays in the charts. This will provide data to add to the model to try to predict track longevity to know which track will be the best investment. Having this prediction, we can also measure revenue per performance and thus help A&R continue to grow. We also suggest getting additional data source, such as new track & chart record data, to expand on model development opportunities. This is with the end goal of being able to predict track success based on different song features, like genres. We understand that these future updates will benefit A&R exponentially and provide better results in the long term.

Our final application is to focus on new artists with no prior charting songs can. We predict whether their songs will chart. See below for tracks released 2017-2019.

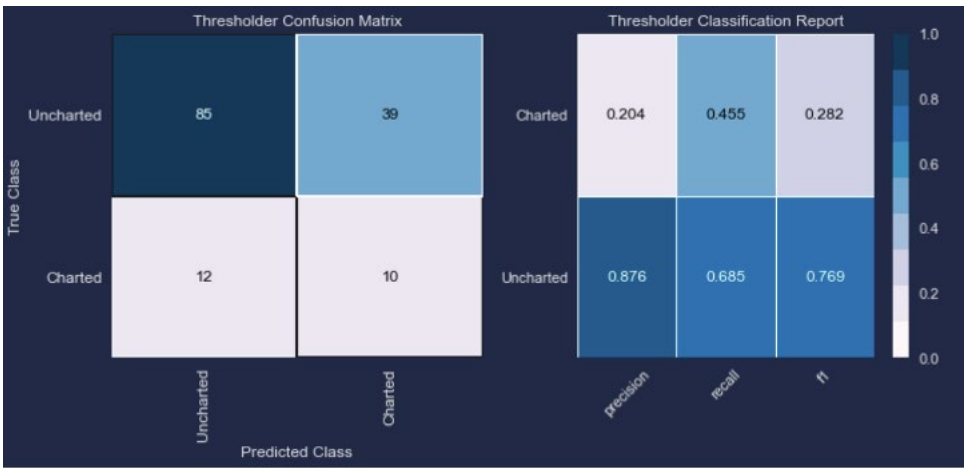

Figure 17: Confusion matrix and classification report of tracks released 2017-2019 by artists who have no previous charting history

We also look to see if the artist will have commercial success later in their career, outside of that single track:

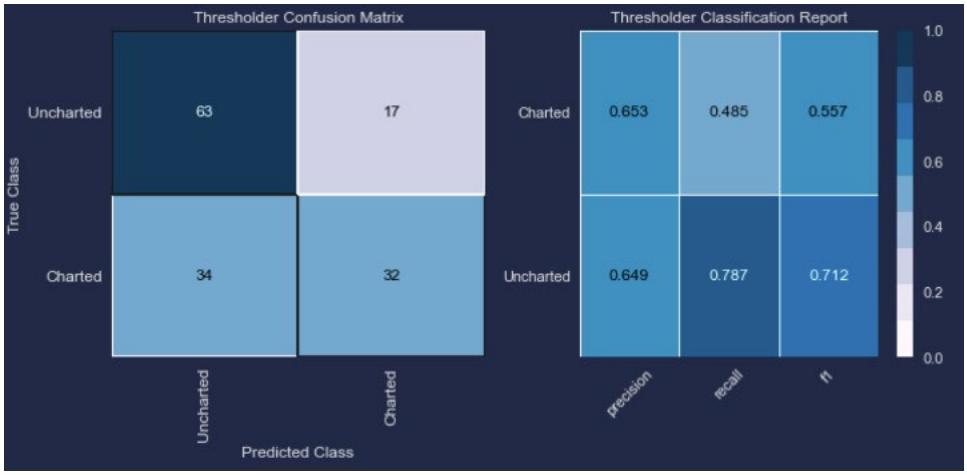

Figure 18: Confusion matrix and classification report of tracks released 2017-2019 by artists who have no previous charting history, using artist lifetime chart months as the label

Although the model may not predict correctly on individual tracks, this was always a challenging task. The model, however, does tend to predict tracks by artists that are worthwhile investing in. Future work should focus on artists as the unit of analysis, rather than tracks, as it appears that the model is better able to predict artist performance, and because it is very rare for a track to chart, due to a variety of reasons.

	artists	name	track_chart_months	artist_chart_months	artist_lifetime_chart_months
584	Juice WRLD	Lucid Dreams	10.0	0.0	14.0
592	DaBaby	BOP	5.0	0.0	28.0
602	City Girls	Act Up	3.0	0.0	3.0
612	Gabby Barrett	I Hope	0.0	0.0	7.0
641	Luke Combs	She Got the Best of Me	2.0	0.0	22.0
652	Arizona Zervas	ROXANNE	0.0	0.0	6.0
688	Lizzo	Boys	0.0	0.0	16.0
694	Lil Yachty, Migos	Peek A Boo	0.0	0.0	13.0
702	Tyler, The Creator	I THINK	0.0	0.0	2.0
753	Lauv	Superhero	0.0	0.0	1.0
754	French Montana, Swae Lee	Unforgettable	0.0	0.0	9.0
770	blackbear	playboy shit (feat. lil aaron)	0.0	0.0	8.0
780	Logic, ROZES	All of Me (feat. Logic, ROZES)	0.0	0.0	5.0
823	DaBaby	INTRO	1.0	0.0	28.0

Figure 19: Selection of tracks released 2017-2019 by artists who have no previous charting history. The table shows artist names, track name, track chart months, artist's previous charting history, and artist's lifetime charting months

There is evidence of bad data, either from Chart2000 or matching charting songs to Spotify. Lil Nas X's *Old Town Town* received a lot of radio play, yet the listed number of charting months is zero. And surely have Lizzo, Migos, Portual. The. Man., Doja Cat, Kodak Black received much radio play, yet artist charting history is similarly missing. This indicates that using the Top 50 Global Charts from Chart2000 is too restrictive. It's common for an artist to chart on genre-based charts, e.g., Billboard Alternative and Billboard Hip-Hop charts, but not appear on the Billboard Hot 100 (which appears to focus exclusively on songs to play at the gym). It's possible that Chart2000 only focuses on the top hits chart, and not genre charts. Thus, it's prudent to add a greater variety of charts and deeper charts.

The team saved all of their work to the following repository: https://github.com/pezon/music-mining.

[i] https://www.musicbusinessworldwide.com/over-60000-tracks-are-now-uploaded-to-spotify-daily-thats-nearly-one-per-second/
[ii] https://www.rollingstone.com/pro/features/are-the-major-record-companies-signing-too-many-artists-847697/
[iii] https://www.kaggle.com/general/232036
[iv] https://chart2000.com/data/chart2000-songmonth-0-3-0063.csv