How to Have a Number One Hit the Easy Way: Discussion

1. ***What was the time range used for data collection? This might be hard to do, but I was wondering if the dataset is across multiple years, was the concept of varying tastes across years taken into account or thought about?***

   Date range for tracks in the dataset: 2000-2021. We're limited to Chart2000, which is the data source for charting statistics. (Chart2000's own data source go back a hundred years.)

   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. [1]

   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.

   Short answer: accounting for varying tastes wasn't a strict consideration, but as we're working with a dataset of music consumed worldwide, the models would likely not home in on any strong taste trends. Instead, the model ideally is generalizing on structural ideas of music that transcend region and decade. Varying tastes is expected to be acceptable model variance.

   [1] "Appetite for Self-Destruction: The Spectacular Crash of the Record Industry in the Digital Age," Steve Knopper. 2009.

2. ***How did you do feature selection from different datasets? What was the criteria? How big was the initial dataset? Looking at the feature results, have you maybe tried to reduce or get rid of seasonality?***

   Feature selection from the source datasets was a matter of domain knowledge and business requirements.

   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 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. As mentioned in the presentation 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.)

The initial dataset was compiled from recordings nominated for a Grammy (n=~535), tracks that have entered the Chart2000 global aggregated monthly song chart (n=~3300). We added random samples from Spotify daily charts (n=~15000). The dataset is de-duplicated on track id and limited to the years 2000-2021. The final dataset is 14,967 tracks and 69 columns.

Also important was using 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, but there were some features which were not complete (e.g., artist gender, country, recording label), so we did not use them.

Seasonality is a really interesting question and would probably be a part of our next steps. It was not within scope of this project, and I don't believe our release date data was specific enough to perform this analysis. We would probably need to collect additional data.

3. ***How did you perform the ranking of importance of features? Which Ensemble Models did you perform? How were some of the audio features (e.g. "danceability", "speechiness") measured in a tangible way? Besides RFE, were there other feature selection methods you tested.***

Methods used for feature selection:

- Recursive feature extraction (with a Random Forest classifier): https://www.scikit-yb.org/en/latest/api/model_selection/rfecv.html
- Step-wise forward/backward selection on the p-value: https://www.datasklr.com/ols-least-squares-regression/variable-selection
- Parallel coordinates plots: https://www.scikit-yb.org/en/latest/api/features/pcoords.html

Ensemble models:

- Random Forest
- Voting classifier with Multi-layer Perceptron, SVC and Random Forest; both ensemble models with soft-voting and hard-voting.

Audio features were compiled by Spotify and EchoNest. See for more information:

- https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features
- https://dl.acm.org/doi/10.1145/1743384.1743428
- https://medium.com/@boplantinga/what-do-spotifys-audio-features-tell-us-about-this-year-s-eurovision-song-contest-66ad188e112a#:~:text=Spotify%20Audio%20Analysis&text=A%20value%20of%200.0%20is,fast%2C%20loud%2C%20and%20noisy.

4. **How do you know you have collected enough data for your project since you are creating your own dataset?** ***What area of those artists and genres that you collect, do you have a focus? and does your data contain all genres? How do you balance your model for overfitting and still have good results?***

It's difficult to know when "enough" of a dataset has been collected. The north star is the business case. We know there's been a shift in the music industry since 2000, and our global chart aggregator data source only has data available from 2000, so that was a limitation on our collection.

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 include songs that have not charted, we sampled randomly from Spotify's Daily Charts. Although this chart is biased, it does serve a purpose. Generally, music consumption has a very long tail, as much songs have under a hundred plays. We wanted to include songs that were somewhat popular.

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 any 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.

To measure overfitting, we used the plots listed 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.

- Learning curve plot: https://www.scikit-yb.org/en/latest/api/model_selection/learning_curve.html
- Validation curve: https://www.scikit-yb.org/en/latest/api/model_selection/validation_curve.html

5. ***Where did you get Spotify's dataset? From public sources or from Spotify's API? Did you analyze the lyrics as one of the important features in terms of a song's popularity?***

We used a Spotify API to get our data and combined it with data from MusicBrainz and Chart2000. We were not able to conduct a complete test of song lyrics. The focus was on the audio features of a song.

To construct a set of track IDs, we randomly sampled songs from Spotify's Daily Charts, collected songs from Chart2000 and used Spotify Search API to retrieve track IDs, collected songs from MusicBrainz's GRAMMY release groups, and also matched those with Spotify with the Search API. We randomly sampled from Spotify's Daily Charts to ensure a class balance of 20%. After compiling a list of track IDs, we used Spotify's bulk APIs to retrieve metadata and audio features.

- Spotify Daily Charts: https://spotifycharts.com/regional/us/daily/latest
- Spotify Track Metadata API: https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-tracks

- Spotify Track Audio Features API: https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features

For more information about collecting and generating the dataset, see the project Github: https://github.com/pezon/music-mining

6. ***How are audio features being processed?***

   The audio features were part of the Spotify dataset. We did not create or process any of these. Audio features were compiled by Spotify and EchoNest. See for more information:

   - https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features
   - https://dl.acm.org/doi/10.1145/1743384.1743428
   - https://medium.com/@boplantinga/what-do-spotifys-audio-features-tell-us-about-this-year-s-eurovision-song-contest-66ad188e112a#:~:text=Spotify%20Audio%20Analysis&text=A%20value%20of%200.0%20is,fast%2C%20loud%2C%20and%20noisy.

   We did scale them when appropriate for the given model being evaluated.

7. ***When you compare model performance using the ROC curve, how you evaluate and make decision when two models' curves cross each other at some points?***

   When ROC curves cross it means models are performing differently at different classification thresholds. We tried to maximize precision while also making sure that we ended up with a reasonable subset of tracks to be evaluated. ROC curves were just one of many tools we used to evaluate our models.

8. ***Could you elaborate on what Fuzzy match search was used for?***

   Fuzzy matching is a technique to match two strings. 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 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 used thefuzz library: https://github.com/seatgeek/thefuzz

   Details about how thefuzz works: https://python.plainenglish.io/all-the-fuzzyness-of-python-72d12d094195

To see an example of how this was employed, see this notebook: https://github.com/pezon/music-mining/blob/master/notebooks/collection/prepare_charts.ipynb

9. ***How can you improve your model if some of important features of the model is not that related to the business case you want to address?***
   The intended goal of the business was to identify which combination of features produce the best performing model. We concluded that a mix of audio features, artist releases and distances to charting songs were the optimal features in support of our business case.

   To improve performance, we have to focus on minimizing Type I or Type II error — in our case, we minimized Type I error, which is time wasted in investing in songs that don't chart, due to limited resources at A&R. The number of songs predicted to our model is relatively small, but acceptable when considering that 60,000 songs are uploaded to Spotify daily. A 50% precision rate is perfectly fine, considering that a model that is not tuned for precision may have a 10% precision rate, and the costs of time and resources and invest in songs that won't chart is much higher.

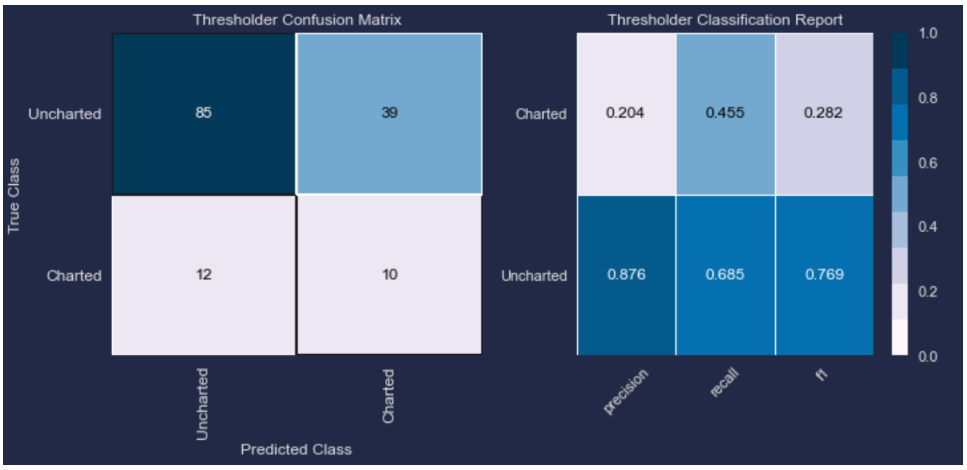
   We've determined that our model underfits, and our options are to include more features and more data from more and deeper charts. Due to resource limitations, we didn't pursue this, but these are recommended next steps.

10. ***Are you focusing on artists who have no debuted or not been featured as a top hit? Did you exclude artists that already have a best hit within an album? I would assume that these would be outliers to your dataset.***

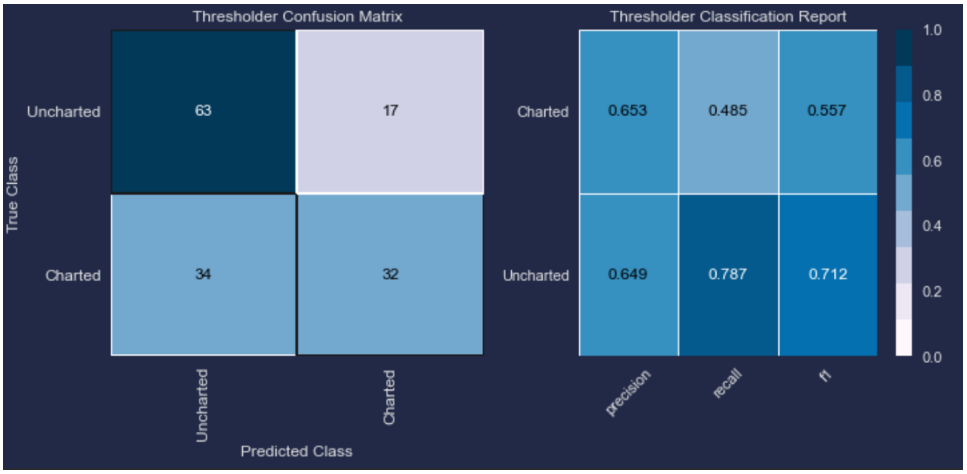
    The goal is to identify artists who have no previous releases, or have not previously charted, and that's why it was important to focus only on audio features; because artist features (such as artist's previous charting history) were such strong indicators of whether a track would chart. An A&R agency is interested on finding new artists, the next big sound.

    In terms of outliers, songs that have charted are outliers. Between 2000 and 2021, only 5000 tracks have graced the top 50 global charts. That represents 21.5% of our dataset. However, our dataset represents tracks that are popular. Spotify has a database of 82 million tracks, and 60,000 tracks are uploaded to Spotify daily. The average song has on average less than a hundred plays. Songs that have charted are the real outlier.

    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.



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



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.

	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

11. How big was the data and how long did it take for you to collect the data?

Just under 15,000 records, 69 columns, and represents tracks released between 2000-2021. The gzipped-compressed CSV file is 3.1 MB. 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.

12. ***How big of an impact did the charting time have on overall revenue generated by the song?***

    Revenue is based on a combination of position on the global chart, the number of months on the chart, and the country of the chart (e.g., US Billboard, UK Top Hits), and the total music industry revenue share of the country in that year, according to the International Federation of the Phonographic Industry (IFPI), and accounting for inflation. Calculating this indicative revenue was performed by Chart2000, and you can read their methodology here: https://chart2000.com/about.htm

13. ***You said the random forest is the best model you are applying. Why is that? How is it supported?***
    In general, the higher on the chart, the longer number of months on the chart, and the more prevalent on US and UK charts, the more revenue generated by the song. Random Forest models had the highest scores on our performance metrics of greatest interest.

14. ***One dataset to contain the whole music genre, do you think of focusing just on the major genre? Does the past number one hit belong to the different genre or the major ones?***

    Genre was not used in our analysis. The final models did not factor in genres. Training models on top-level genres is listed as recommended next steps. It may produce a more precise model.

    Spotify and MusicBrainz both attribute genres on the artist profile, and there are multiple genres attributed to artists. There are thousands of genres, sub-genres, derivative genres, etc., and many ways to classify genres, as they exist as marketing labels and music journalism terms. MusicBrainz (and Last.FM) may have tagging on the recording level, but in general crowd sourced genre tagging is fairly unreliable and difficult to work with.

    Our approach to genres is to construct a TF-IDF of n-grams of genres attributed to artists (I.e., “french hip hop” -> “french,” “hip,” “hop,” “french hip,” “hip hop,” etc.), selecting the top 25 genre n-grams by frequency, and assigning a single genre, the lowest frequency genre attributed to an artist (so that “teen pop” is assigned before “pop,” because “teen pop” is more precise); otherwise, the genre is “other.”

15. ***One dataset to contain the whole music genre, do you think of focusing just on the major genre?***
    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.

The final models did not factor in genres. Training models on top-level genres is listed as recommended next steps. It may produce a more precise model, for example, to predict whether a song will perform well on the EDM charts, and filter the training dataset on EDM songs. It would be prudent to also use EDM-focused charts from EDM data sources; currently, we use global charts, which are not focused on any genre.

Spotify and MusicBrainz both attribute genres on the artist profile, and there are multiple genres attributed to artists. There are thousands of genres, sub-genres, derivative genres, etc., and many ways to classify genres, as they exist as marketing labels and music journalism terms. MusicBrainz (and Last.FM) may have tagging on the recording level, but in general crowd sourced genre tagging is fairly unreliable and difficult to work with.

Our approach to genres is to construct a TF-IDF of n-grams of genres attributed to artists (I.e., “french hip hop” -> “french,” “hip,” “hop,” “french hip,” “hip hop,” etc.), selecting the top 25 genre n-grams by frequency, and assigning a single genre, the lowest frequency genre attributed to an artist (so that “teen pop” is assigned before “pop,” because “teen pop” is more precise); otherwise, the genre is “other.”

16. ***Choosing Spotify seems a smart pick, but is the choice of one app represent the whole music industry. Since Spotify is not used for the whole world, are you generally focus the data on the U.S. level?***

In our case, 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 their methodology here: https://chart2000.com/about.htm

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 absolutely a 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 $8 bn, or 34.6% of the recording industry total revenue of 23.1 bn, by far the largest market.

17. ***What was the metric used to measure success in classification?***

    Whether a track charted for at least one month. In our dataset, this variable is a count variable called "track_chart_months," and it is based on data from Chart2000, which aggregates global song charts and produces a top 50 list on a *monthly* basis. So, the meaning of "success" depends on Chart2000's methodology for producing their list, and the fact that we're limited to the top 50 songs in a month.

    We focused on the AUC and precision to evaluate model performance. Focusing on performance allows us to minimize Type I error. We determined that in our model, Type I errors were artists that would be pursued for investment, but the artist's track would not chart. The cost of a Type I error is time, resources and capital. Type II errors are cases where we missed an artist that produced a charting track. The cost of the error is revenue opportunity and reputation.

18. ***Was there any hypothesis around why audio data features weren't working well? For the prediction, is it going to predict the artist or the music genre?***

    The hypothesis is that the music industry is very political, and the popularity and charting position of a track was based on recording labels and publishers' marketing budgets.

    As an example, in their heyday, labels would produce and move massive amounts of units in anticipation of selling large volumes of a single, EP or LP, which positively inflated the revenue generated of a single, its radio play, and Billboard standing. It was a positive feedback loop.

    There is no direct way to measure politics, and we don't have a dataset of marketing budgets of labels, publishers and distributors. However, by looking at the feature importance of all the features – audio and artist features — we showed that the variance explained by audio features compared to other features is relatively small. The most important feature is "artist_chart_months" followed by "artist_chart_tracks," which are the number of total months the artist had previously had a track in the top 50 charts, and the number of tracks an artist has had charted, respectively.

    That having been said, this is not a direct hypothesis test, and there could be other reasons why our model was unable perform well on audio features alone. We also trained an SVM model with a polynomial kernel on audio features alone, and that model performed poorly as well.

19. ***Why was precision used over recall or accuracy score?***

We determined that in our model, Type I errors were artists that would be pursued for investment, but the artist's track would not chart. The cost of a Type I error is time, resources and capital. Type II errors are cases where we missed an artist that produced a charting track. The cost of the error is revenue opportunity and reputation.

A&R agencies today have limited resources (due to collapsing music industry revenue and budgets following the disruption in the 2000s). There are 60,000 tracks uploaded to Spotify **daily**. We do not expect an A&R agency to have resources and budget to comb through tens of thousands of songs, therefore, we are providing a small set of tracks that have a high probability of charting. By optimizing precision, we minimize Type I error, and we produce a small set of tracks, catering to an A&R agency's limited time and resources.

For example, out of 5000 songs, we'll select 200 tracks, and out of those tracks we are confident that 55% of them will chart for at least one month.

20. What is the final model you have chosen?

After looking at all the models performance, taking into consideration latest results of fine tuning and threshold checking, we understand that the model with the best performance metrics is the one to choose, which is the Random Forest, which had highest AUC, Precision and F1 results.

21. More information and bigger images.

For more information about the dataset, the code, and graphics, check out the public github: https://github.com/pezon/music-mining

For bigger images shown in the presentation related to data collection: https://github.com/pezon/music-mining/blob/master/presentation/Pipeline-DataCollection.png

For bigger image shown in the presentation about modeling pipeline: https://github.com/pezon/music-mining/blob/master/presentation/Pipeline-Modeling.png

For bigger images related to EDA, feature selection and feature importance, see the EDA notebook: https://github.com/pezon/music-mining/blob/master/notebooks/analysis/EDA.ipynb

For bigger images related to modeling results, see the Modeling Selection notebook: https://github.com/pezon/music-mining/blob/master/notebooks/analysis/Model%20Selection.ipynb

For the presentation PPT: https://github.com/pezon/music-mining/blob/master/presentation/Music_Presentation.pptx