Catching the Earworm: Understanding Streaming Music Popularity Using Machine Learning Models

Andrea Gao[1]

[1]Shanghai High School International Division Shanghai, China

Abstract—The digitization of music has fundamentally changed the consumption patterns of music, such that the music popularity has been redefined in the streaming era. Still, production of hit music that capture the lion's share of music consumption remains the central focus of business operations in music industry. This paper investigates the underlying mechanism that drives music popularity on a popular streaming platform. This research uses machine learning models to examine the predictability of music popularity in terms of its embedded information: audio features and artists. This paper further complements the predictive model by introducing interpretable model to identify what features significantly affect popularity. Results find that machine learning models are capable of making highly accurate predictions, suggesting opportunities in music production that could largely increase possibility of success within streaming era. The findings have important economic implications for music industry to produce and promote the music using controllable and predictable features tailored to the consumer's preferences.

1 INTRODUCTION

In the past century music production was mainly in the form of physical albums such as cassette tape and CDs. Singers and music producers record a list of song tracks to sell as a whole set of physical album. As such, artists and music industry heavily focus on maximizing album sales, which was one of the most critical measures of artists' popularity and business success [1]. Economically speaking, music can be considered as very similar type of goods like foods, movies, books. As an experience good, consumers don't know the value and quality of the music before purchasing. To resolve the uncertainty associated, consumers need to rely on other information – such as the music's genre, fame of the artist, the album's total sales volume, and the music chart – as reference to make one's choice. At the same time, consumers have to pay a high search cost for the track they actually want since one album typically contains a curated list of tracks. To make the best utilization of money, consumers tend to rely even more heavily on the "other information" e.g., to leverage critic's rating and word-of-mouth popularity of the album, taking wisdom of the crowd when selecting music to purchase. Therefore, hit albums that have already accrued high sales figures will receive further more sales. In other words, the sales of new music albums from popular artists may be amplified by the indirect information, but not the quality of the music. This leads to a significant divide between artists who may produce the music about the same level but end up with vastly different levels of success.

The 21th century has witnessed the technological advancement in music industry that allowed consumers to store music in hard disks such as MP3 or iPods. The increasing prevalence of smart phones and the digitization of music prompted the establishment and wide usage of numerous music-listening apps such as Spotify, Google Play Music and Apple Music, among others, that gradually replaced CDs. Such switch of music consumptions, from purchasing physical albums to purchasing the single track, not only changed the customer experience, but also fundamentally changed the economics of the music industry. For example, Apple pioneered the sales channel through its iTunes store since 2001, a digital platform that sells all tracks at $0.99. Consumers become much free to choose which track to listen rather than purchasing an album with the whole set of tracks, which is now independent of the price effect.

Due to such a music industry evolution, Chris Anderson (2004) proposed the long tail theory to characterize the music consumption in digital era, in which a large portion of tracks that were once unknown have gained certain level of popularity altogether to form a long tail of the consumption distribution. This implies that the popularity of the music and artists may spread within a larger range, increasing sales of less known tracks from nearly zero to few.

More recently, the emergence of streaming platform designs such as Netease Music, QQ Music, Pandora, Spotify, as well as the utilization of Artificial Intelligence into music recommendations have gradually exhibited a spill-over effect [2] – music listened by other users with similar histories are recommended, thus increasing the music popularity as it spreads from several users to a larger group. This pushed a short list of tracks to become uniquely popular. In 2018, Professor Serguei Netessine from

Wharton University of Pennsylvania stated in his podcast that, “We found that, if anything, you see more and more concentration of demand at the top”. Although the podcast focused on movie sales, experiences goods like theater and music sales occur in a similar fashion as shown in data distribution demonstrated in Section 3B. In the book “All you need to know about the music industry” by Passman, he highlighted key differences between music business in the streaming era and record sales. [3] In the days of record sales, artists get paid the same money for each record sold, regardless of whether a buyer listened to it once or a thousand times. But today, the more listens the music tracks have, the more money the artists make. Meanwhile, records sales do not have strong spillover effects as fans of different artists/genres will purchase what they like anyway. In fact, a hit album would bring a lot of people into record stores, and that increased the chances of selling other records. But in the streaming world, that’s no longer true. The more listens one artist gets, the less money other artists would make. In other words, the music consumption is undertaking a radical shift which may affect the definition of popularity in the streaming era, however, it is yet severely underexplored.

Inspired by the evolution of music industry in the recent decades and the recent debunk of long tail theory given a high concentration of popularity for a short list of tracks, this paper aims to investigate the popularity of music tracks on streaming platform, largely different and not extensively explored about compared to that measured by album sales, when it is impacted by the consumer choices and prevalent recommendation features. In particular, rather than considering the level of advertisement, the inclusion in playlists of Billboard Hot 100 as Luis Aguiar and Joel Waldfogel have noted, this paper focuses on leveraging music tracks’ inherent audio features – such as the key, loudness, tempo or measure of emotions to discover the underlying patterns that drive the popularity. [4] Based on the concept of earworms [5] – catchy pieces of music that repeats through a person’s mind – come from these tracks with uniquely high popularity, we hypothesize that audio features may play an important role in determining popularity. According to Seabrook (2016), the music industry has transformed the business to produce tracks that have everyone hooked, through various means including marketing, technology and even the way the human brains work.[1] [6] For example, Jakubowski et al. discussed whether earworms have certain common-grounds or are composed according to certain formulas in a psychological lens, which gives inspirations of this paper to study the similar subject matter from the lens of machine learning. [7] As streaming, or the number of listens, becomes the main measure method of popularity, the cost of listening to music further decreases from individual songs to nearly none on some platforms as long as consumers have gained premier. Consumer’s freedom further increases and may listen to a song due to interest and curiosity, and many todays believe that popularity is gained from information in social media [8]. Although many of other factors such as social media play a role in determining music popularity, this paper provides an additional perspective to the existing psychological or platform studies, and provides suggestions for the most controllable element in music production – composition. The results of this paper hope to give implications upon how music in the current digital age, which popularity is measured by streaming volume, can be composed to gain the large volumes and thereby create economic value. Whether music in current ages have become more predictable due to platform economics or the hope of producing an earworm is an important economic topic of interest that may cause large changes in the music production industry.

In this paper, we aim to predict the popularity of tracks based on acoustic features, together with historic performance of artist. In particular, this paper includes several state-of-the-art machine learning models that give relatively implications of what features to what extent could make tracks become popular. Using a large-scale dataset, which incorporates more than 130,000 tracks from a world-leading music streaming platform Spotify, we adopted various advanced machine learning models to intelligently classify which song tracks would have high popularity based on comprehensive audio features derived from each track. This adds to the recent works that tried to predict the music genres with audio features [9], in which the machine learning models are used to categorize music genres. In particular, we further introduced an explainable AI tool as SHAP [10] to capture how these audio features have differential impact on the track popularity. As such, the presented results may provide strong implications to understanding the popularity in the streaming music platform and mechanism of essential audio features that music industry may consider for production.

The paper is presented as the following. The second section discusses previous works that analyzed the economics of music industry and determinants of music popularity. The third section introduces the Spotify dataset such as its features, categorical data assignments, and general explorative data analysis demonstrations. The fourth section explains several machine learning models used in this work in detail, – including Multiple Linear Regression, Logistic Regression, Decision Tree, Random Forest, and Boosting tree, and Neural Networks – the mechanism of the models, as well as the hyper-parameters used. Next, the fifth section presents the prediction results, evaluates model performances, and interprets feature importance. Last, we discuss the implications of the research and conclude.

2 Related Works

2.1 Economics of Music

In 2005, Marie Connolly and Alan B. Krueger, two researchers from Princeton University, investigated the economics of the rock and roll industry. [11] They were of the first researchers who implemented economic knowledge to study the subject matter of music. Connoly and Krueger observed changes in the rock and roll concert industry during 1990s, and explained how these changes have contributed to factors like concentration of revenue among performers and copyright protection. [1] As the research was mostly descriptive and lies on the social

[1] Seabrook (2016) showed some anecdotal evidences that a company named Hit Song Science used a computer-based method of hit prediction that purported to analyze the acoustic properties and underlying mathematical patterns of the past hits.

science research by concluding from massive survey results, the results, in the form of analysis, were valuable for studies but were neither numerical nor quantifiable.

In the more recent decade, an article named "Digital music consumption on the Internet: Evidence from clickstream data" by Aguiar et al. [12] focused on the effect of digital era and streaming on digital music sales. The authors stated the "stimulating effect of music streaming on digital music sales" and that purchasing behaviors of customers have changed since 2000s due to the digital music platforms, largely statistical and economic. Music industry revolution was also discussed in Hubert Léveillé Gauvin that studied the compositional practices of popular music (number of words in title, main tempo, time before the voice enters, time before the title is mentioned, and self-focus in lyrical content) and its changes with hundreds of songs from U.S. top-10 singles in 1986~2015. [13] It shows that popular music composition indeed evolves toward grabbing listeners' attention, consistent with the "attention economy" it proposes that the music industry has already transformed into.

In this paper, we build upon Aguiar and Martens's conclusion about the effects of digitization on platforms and investigate to what extent has this phenomenon caused people to have similar interests in music by looking specifically at most popular music. [12] This allows us to testify some of the previous works, about whether music interests have been converging due to music platforms' changes. Furthermore, instead of statistical analysis as in previous works, we utilize machine learning models (logistic regression, random forest, neural networks, etc.) to investigate trends in a large collection of diverse music tracks.

In the article "Platforms, Promotion, and Product Discovery: Evidence from Spotify Playlists", Aguiar and Waldfogel analyzed streaming platform and promotion techniques' effects on music sales. [4] They specifically compared the difference in streaming volumes of tracks before and after they were added to global playlists as well as when tracks just entered and just left the top 50 popular list. They also aggregated the streaming volumes to observe effects of playlists on tracks' popularity. The effects were large and significant: adding the track to a popular list called "Today's Top Hits" may increase the stream volumes of tracks by "almost 20 million that worth between $116,000 and $163,000". Furthermore, inclusion in "New Music Friday lists" generally raised possibility of success for tracks, regardless of the artist's original popularity. Therefore, not only showing how platform economics such as streaming and playlists might have caused large effects on music popularity, this research further implies the huge economic value of successful tracks, which increases the value of this paper's purpose and results. In an economic lens, the predictability of music popularity in terms of audio features, the most fundamental and controllable elements of a song, is highly valuable for the industry. [14]

2.2 Determinants of Music Popularity

Some research in the more recent years have been exploring similar topics – music popularity. Matthew J. Salganik, Peter Sheridan Dodds, and Duncan J. Watts investigated the predictability of songs, stating that most successful are of high quality, most of the low-quality songs are unpopular, leaving the rest in between to be unpredictable. [15] Moreover, Kelsey McKinney discussed the history of track lengths' changes in the recent decade, and concluded that song lengths between 3 to 5 minutes are more likely to become hit songs. [16]

Several papers have used machine learning models to explain popularity by features. Askin et al. utilized computer science tools to analyze the musical features of nearly 27,000 tracks – including those in Hot 100 – and concluded that tracks that are less similar to their peers are more likely to succeed, to some extent disproving the claim that popular music all sound alike and that earworms can be composed through certain formula. [17] This gives some implication to the modeling process in this paper that linear models between features and popularity may not fit well, may require more complex models that capture the non-linearity. Herremans et al. used basic musical features and several classification models to gain insight in how to produce dance hit track, aiming to provide benefits towards the music industry. While their work gives some insights to how popularity can be predicted and to what extent hit tracks can be produced, the models used in their work (ie. Logistic Regression, C4.5 Tree, Support Vector Machine Classifiers) were relatively simple and basic, in which the features are not clearly explained. [14]

Furthermore, Araujo et al., built upon the work by Herremans, et al to predict whether a song would enter Spotify's Top 50 Global ranking using classification models. [14,19] It takes the platform's previous Top 50 Global ranking as well as acoustic features into account and built upon several models, including Ada Boost and Random Forests, probabilistic models with Bernoulli and Gaussian Naive Bayes, and multiple kernels for the Support Vector Machine classifier. Likewise, Interiano et al. analyzed more than 500,000 songs released in UK between 1985 and 2015 to understand the dynamics of success, by correlating acoustic features and the successes in terms of official chart top 100. [14] Their work also showed the acoustic features has high predictability of the success.

3 Data

3.1 Spotify Dataset

In the Spotify dataset, there are 130,663 tracks in total that was collected in 2018 and 2019, respectively. For each track, there are 14 numerical values in addition to the track's name and the artist. 13 of the numerical values are audio features, and the other is the label in this paper, popularity.

3.1.1 Feature Explanation

TABLE I. SPOTIFY DATASET FEATURE EXPLANATION

Variable	Description	Mean	Std.
Dependent Var.			
popularity	Overall estimated amount of streaming	24.209	19.713
Audio Features			
duration_ms	Duration of the track in milliseconds.	212633.1	123155.1
key	Estimated overall key of the track using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on.	5.232	3.603
Mode	Modality (major or minor) of a track. Major is marked as 1, and minor is marked as 0.	0.608	0.488
time_signature	Estimated overall time signature (notational convention about how many beats in each bar).	3.879	0.514
acousticness	Confidence measure of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.	0.343	0.346
danceability	Suitability of a track for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.	0.581	0.190
energy	Perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.	0.569	0.260
instrumentalness	Measure of whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.	0.224	0.360
liveness	Measure in 0~1 of presence of an audience. Higher liveness values represent an increased probability that the track was performed live.	0.195	0.168
loudness	Measure of average loudness of a track in decibels (dB).	-9.974	6.544
speechiness	Measure of presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.	0.112	0.124
valence	Measure of musical positiveness (ie. happy) in 0~1. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while low valence means tracks are more negative (e.g. sad, depressed, angry).	0.440	0.259
tempo	Estimated tempo of a track in beats per minute.	119.473	30.160
Total Observations	130663		

Notes: In this table, "duration_ms", "key", "mode", "time_signature", "loudness", and "tempo" are information directly extracted from the tracks. Also, "acousticness", "danceability", "energy", "instrumentalness", "liveness", "speechiness", and "valence" are values defined and calculated by Spotify according to certain internal algorithm.
Source: Spotify Dataset[2]

3.1.2 Measurement of Popularity

According to Spotify's website for developers, the popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by Spotify's internal algorithm and is based on the total number of plays of the track and how recent those plays are. In other words, the popularity is based on both the volume and regency of the streams.

Generally speaking, tracks that are being played a lot recently will have a higher popularity than tracks that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity are derived mathematically from track popularity.

Given that track popularity can be directly measured at the streaming platform as the function of number of listens, in comparison to the past times, when music was sold as physical albums. For example, the presence of pirated albums as well as the unknown number of listens makes the official number of album sales inaccurate for the real popularity. This is one significant advantage of digitization which provides much more accurate measurements of popularity than the old days.

3.2 Explorative Data Analysis and Data Standardization

In explorative data analysis, we used histograms of features, a correlation heat map of all features included, as well as bar graphs and scatter plots with simple regression lines of several features with popularity.

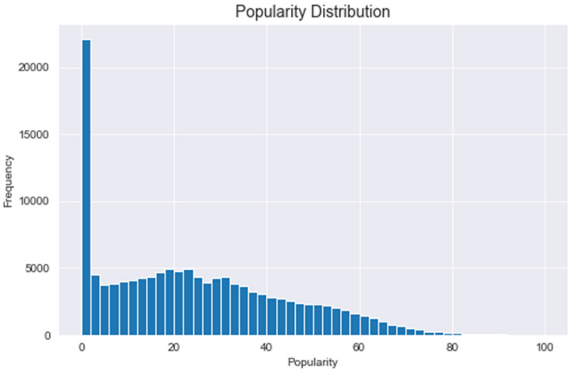

Figure 1. Visualizer Popularity Distribution Before Scaling

This appears to support the long tail theory on first glance, as a large majority of the tracks have extremely low popularities at about zero and a nearly unseen portion of tracks have high popularity between 80 and 100. To make the results more valuable, we include only songs that are at least listened to an accountable number of times, we excluded the songs with popularity of 0.

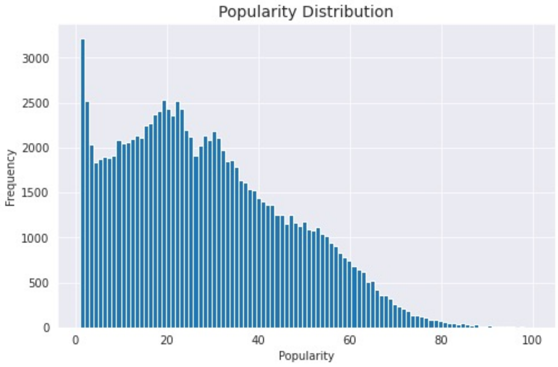

Figure 2. Popularity Distribution, After Removal of Zero Popularity Tracks

After excluding tracks with popularity of 0, the tracks' popularity distribution visually differs from what is stated by long tail theory, or a strictly inverse relationship between popularity and the number of songs with that popularity. In this graph, there are instances where there are increasing number of songs with increasing popularity, called a chasm.

As we acknowledge that there is a large range of tracks that have extremely low popularity, even zero according to Spotify's calculation algorithm, it is not sensible to study this as a regression problem and predict specific popularity of songs. As seen in this Figure, the chasms serve as thresholds, above which the number of tracks with higher popularity continuously decreases. Most songs that cannot pass such a threshold and enter the decreasing trend can rarely be successful. Therefore, rather than predicting specific popularities of songs, we can classify the songs into popular and unpopular. An interesting fact is that the lowest point of the chasm at around popularity of 25 marks exactly 50% of all songs. In other words, exactly 50% of songs have either passed or have popularity below 25. Therefore, as we hope to focus solely on those songs that have succeeded and passed through such threshold that become successful, the labels for classification models in this investigation are set such that the tracks with top 25% in popularity are classified as popular, and the remaining 75% are unpopular. The reason to set 25% as the classification boundary is to ensure that tracks above this level are really those that have succeeded, having passed the chasm with a large extent. The tracks with popularity of zero are removed before classification since they do not receive any interaction with the users. The benchmark was calculated to be at 41, as 25% of the non-zero-popularity tracks have popularity larger or equal to 41. The popular tracks are marked as 1 while the unpopular are marked as 0.

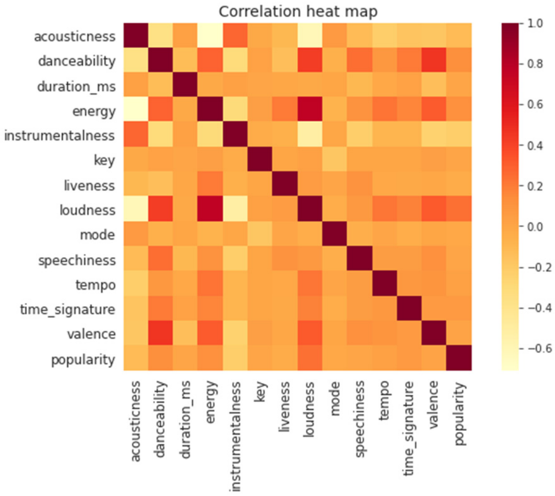

Figure 3. Correlation Heat Map of Spotify Features

The figure above shows a heat map that demonstrates the correlation between features pairwise. The larger direct correlation, the darker colors, or larger heat, of the small square. From this figure, we can grasp a general trend of which features may be correlated. For example, loudness, or the average decibel of a piece, has a close correlation with energy as denoted by the dark red square, while energy and acousticness don't have as strong a correlation as shown by the light-yellow colored squares that is close to white. Note that for loudness, if a piece has occasional large volume or largely fluctuating volume as seen often in symphonic music, which has large dynamics in volume,

[2] More information of Spotify Dataset can be found on this website: https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

then its loudness would be relatively low as Spotify calculates it as an averaged value; if a song is of consistent high volume, then its loudness would be relatively higher.

Focusing on popularity, we can see that the features' correlations with popularity are generally moderate, while instrumentalness has the weakest and loudness has the strongest correlation. This serves as a reference when evaluating models as we have in mind what features may affect popularity more.

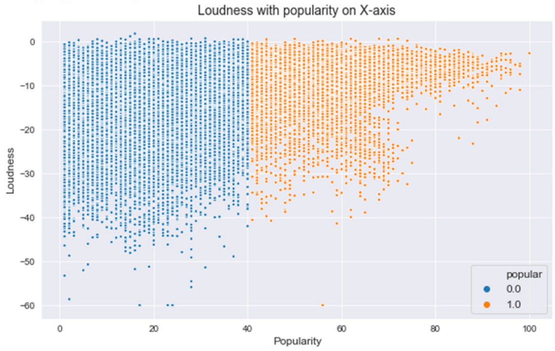

Figure 4. Scatter Plot of Popularity and Loudness

Using the classification method of popular and unpopular tracks in the previous sub-section, the tracks are colored accordingly with orange and blue representing popular and unpopular tracks respectively. According to this graph with popularity on the x-axis and loudness on the y-axis, there is a clear trend that the orange popular data points generally have higher loudness, suggesting possible relationships between the two factors that popular tracks generally are louder overall. According to loudness's definition and its implications, this shows that tracks with large dynamics in loudness might not be as popular as those with more stable and large volume.

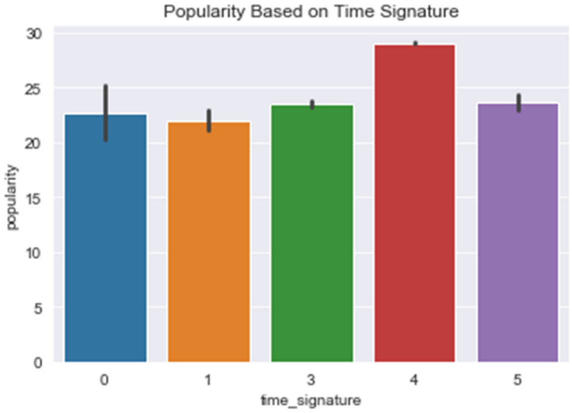

Figure 5. Bar Plot of Average Popularity Based on Time Signature

Above is the graph of the average popularity for tracks with certain time signature, or the number of beats at each bar in a certain piece of music. This bar graph shows that the average popularity of tracks with time signature of 4 is generally higher than tracks of other time signatures. 4 is a very ordinary and commonly used signature used in all kinds of music. On the other hand, only rarely are 5 beats per bar used in popular music because it generally produces an awkward rhythm for listeners. From Figure 5 we can see that the most common and popular 4 beat per bar might be more preferred by music consumers, while those unusual time signatures might naturally sound not as attractive, implying that tracks with time signature of 4 may generally have higher popularity and possibility to succeed.

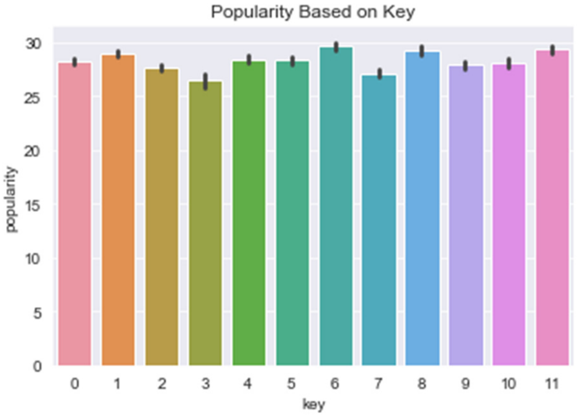

Figure 6. Bar Plot of Average Popularity Based on Key

In the graph above, we see the bar graph of average popularity for tracks with each key. Key is one of the most basic features within the composition of a piece of music. As one of the most fundamental features that can be changed for a track easily by modulating from one key to another, one may expect average popularity of tracks from each key to be on a relatively similar level if they are really unaffecting popularity. However, in this plot, we observe some relatively considerable differences in the average popularities. For example, average popularities of key 6, 8, and 11 are higher than those of keys 3 and 7. This might imply that keys do have certain effects on popularity, which would be further explored later in the Results section.

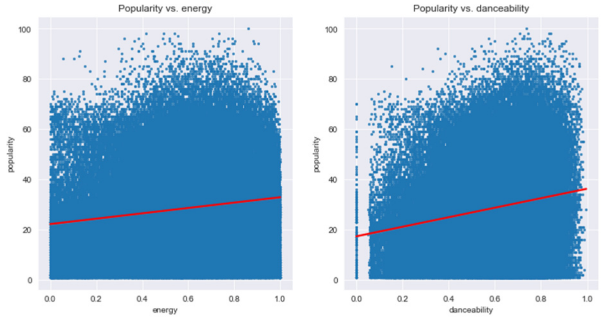

Figure 7. Bar Plot of Average Popularity Based on Key

In the plots of popularity against energy and danceability respectively, both plots suggest that a moderately higher energy and danceability have the highest popularity values, while this trend is clearer between popularity and danceability as low danceability barely has any popular tracks, which is not seen in the plot for popularity and energy. When specifically observing values of energy and danceability of those songs with especially high popularity, larger values of these two features appear to be the majority.

Other than explorative data analysis, we also wanted to use artists' names as a categorical feature. It is normal to recognize that some tracks are famous due to the singer's popularity, while the internal features of the tracks are not as important. Therefore, we divided all singers into 4 groups, according to the number of songs they have in the dataset. Artists of the least 25% number of tracks is labeled "first", 25% to 50% labeled "second", 50% to 75% labeled "third", and the rest labeled "fourth". Thereby, we have a quartile variable that measures whether certain artists are experienced.

We noticed that the number of tracks of artists cannot directly tell the artists' popularity while we expect an artist's popularity to large affect the songs this artist produces in the future. Therefore, we brought in a similar dataset as the Spotify dataset we are mainly using, but from an earlier time – the main dataset is from April 2019, while this similar dataset is collected in November 2018. Note that the Spotify internal algorithm take only streaming in a short time frame into account, these two datasets have tracks with largely different popularity. The number of popular tracks in this similar November 2018 dataset, calculated with similar methods as detailed in previous sections, is counted for each artist, and all artists with at least one popular track are classified into quartiles with an additional category for those artists without any popular tracks. This additional feature marks tracks in the main dataset with artists without any popular songs in the November 2018 dataset as "first", the tracks with artists of bottom 25% in number of popular tracks as "second", etc. until "fifth" for tracks with the artists with top 25% number of popular tracks in the November 2018 dataset.

Other than the plots shown above, the explorative data analysis results as well as visualizations of all features are given in Appendix A.

Finally, for more comparable results and to ensure one feature doesn't affect the dependent variable too much, we scaled all features including popularity to a scale of 0~1.

3.3 Training and Testing Data

After explored some information implied in the data such as correlation between different features and added potentially helpful categorical features, it is important to prepare data to be used in models.

First, as our classification method separates tracks into popular and unpopular ones, the former includes 25% while the later includes 75% of all tracks. This causes the two classes to be unevenly distributed. To solve this problem, we used undersampling to balance the two classes by randomly selecting the same number of samples from negative class as that in positive class. After undersampling, there are 32983 samples in each of the positive and negative classes.

Furthermore, each model is trained on a training set and tested on a testing set. From the 65966 samples in total, 80% are randomly chosen as the training set and the remaining 20% of samples are the testing set. This allows the test the generalizability of models by observing whether the model predicts the label, popularity in this case, successfully as measured by the evaluation metrics. As generalizability is one of the most crucial factors considered in the model, the split of data into training and testing subsets evaluates whether models predict successfully, thereby implying whether models can accurately and realistically show the determinants of tracks being popular.

4 Machine Learning Models

After preparing the dataset with comparable features and certain noticeable correlations, the next step is to train high quality models and continuously improving upon them through hyper-parameter tuning and Principle Component Analysis (PCA) to create the model that produces the best results.

In this paper, multi-variate linear regression, logistic regression, and decision tree are the baseline models on which we hope to utilize different methodologies to improve upon and distinguish the most influential features that make music popularity predictable.

4.1 List of models used in this work

4.1.1 Multi-variate linear Regression

Multi-variate linear regression is one of our baseline models in this paper. It is a simple regression machine learning model that predicts numerical values by giving each feature a weight that denotes how it affects the dependent variable, which is popularity in this case.

Least-squares regression sums the squares of the residuals and finds the coefficients minimize the sum. It prevents the errors with opposite signs from cancelling out, thus creating fairer regression results. [2] For this linear regression, the total sum of squares of the vertical distance between the line and each data point is called the cost function, or least-square error (LSE) function, which is hoped to be minimized. The equation is $\hat{y} = \theta^T \cdot \mathbf{x}$, in which $\hat{y}$ is a vector of all predicted values; θ^T is the transpose of the vector of coefficients in the multivariate regression equation, which is a row vector; and $\mathbf{x}$ is a matrix of all data, including values of all features for each data point. [20] The cost function is

$$J(\theta) = LSE = \sum_{i=1}^{m} (\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)})^2 \qquad (1)$$

To find the θ vector that minimizes this cost function $J(\theta)$, we used the normal equation method to do this work. [20]

Since the value outputs are continuous, we used a consistent classification method as detailed in section 3. Popularity of larger than or equal to 0.41 (scaled) is popular.

4.1.2 Logistic Regression

Logistic regression is another baseline model used in this paper. It is a simple classification model that predicts the possibility of some instance falling into binary classes of 0 or 1, defined in this paper as unpopular or popular tracks. If the possibility calculated is larger than 50%, then the instance is classified as 1; if not, it is classified as 0.

To fit the continuous output values (as in linear regression) within the range of 0~1, the sigmoid function $h_\theta(x) = \sigma(\theta^T x) = \frac{1}{1+e^{(-\theta^T x)}}$ is used. This transfer function bounds the output to a "S" shape curve within 0~1 as shown in the following figure. We interpret the output as a probability predicted from a range of input values represented by $\theta^T x$.

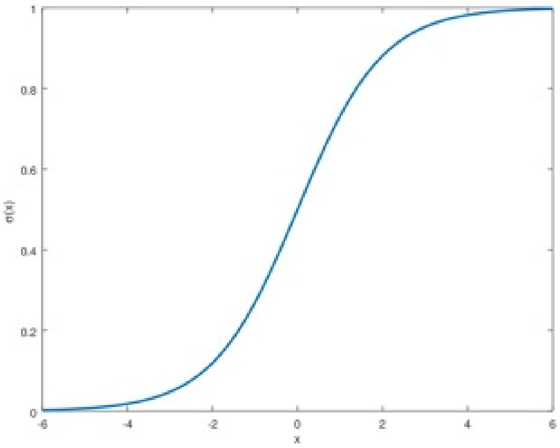

Figure 8. Sigmoid Function[3]

Similar as linear regression, logistic regression also has a cost function that is minimized to find the coefficients θ. In logistic regression, the method is called Maximum Likelihood Estimation. [20] The log loss function is

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} logh_{\theta}\left(x^{(i)}\right)$$
$$+(1-y^{(i)})\log\left(1-h_{\theta}\left(x^{(i)}\right)\right) \qquad (2)$$

in which $y^{(i)}$ is the real label value. The loss function serves its role by increasing in value if the predicted value $h_{\theta}\left(x^{(i)}\right)$ is largely different from $y^{(i)}$. By repeatedly calculating the partial derivative of the loss function, θ_j is updated after each iteration, j denoting the current integration.

$$partial\ derivative\ of\ J(\theta) = \frac{\partial J(\theta)}{\partial \theta_j}$$
$$= \frac{1}{m}\sum_{i=1}^{m}\left(h_{\theta}\left(x^{(i)}\right) - y^{(i)}\right) x_j{}^{(i)} \qquad (3)$$

$$\theta_j = \theta_{j-1} - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_{\theta}\left(x^{(i)}\right) - y^{(i)}\right) x_j{}^{(i)} \qquad (4)$$

in which α represents the learning rate hyper-parameter. If the partial derivative is positive, the loss function is increasing, then next θ_j is reduced; if negative, then increasing the next set of parameters would possibly result in lower loss. This process is repeated until convergence. In scikit-learn, we utilize the built-in Logistic Regression model to employ such a process.

L2 Regularization is used to eliminate outlier weights which may occur during the training process. An additional term $\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j{}^2$ is added to the loss function called the penalty term. This term increases the cost when there are outlier weights, thereby forcing the outlier weights lower to an average level. [20] This increases generalizability of the model.

4.1.3 Decision Tree

Decision tree is a supervised machine learning model for classification or regression usage. In our paper, we use the Decision tree classifier model for consistency. Decision tree, as its name implies, allows the prediction of certain values by following the decisions throughout the tree from the root to the leaf nodes. This stepwise and white box prediction that allows humans to interpret how the results are predicted, which is one of its advantages.

We directly used Decision tree method in ski-learn, which implements the Classification and Regression Trees (CART) algorithm, the latest version until now.

The cost function of decision tree is utilized at every level within the model. The set at each node is split into two subsets using $J(k, t_k) = \frac{m_{left}}{m}G_{left} + \frac{m_{right}}{m}G_{right}$, where left/right denote each of the two subsets, $G_{left/right}$ measures the impurity of the corresponding subset, m is the number of total samples, and $m_{left/right}$ is the number of samples within the corresponding subset. Once the set is split into two after minimizing this cost function, the algorithm continues until the maximum depth is reached, which is a hyper-parameter, or when it cannot continue splitting to reduce impurities. [20]

Note that the decision tree algorithm is a greedy algorithm, meaning that it splits the current set to minimize the current level's impurities but does not consider the impurities several levels down. This may imply that Decision tree is not being the optimal model.

4.1.4 Random Forest

Random forest is a model that uses bagging on multiple CART Decision trees. In other words, random forest builds multiple Decision tree models in parallel and uses "majority votes" to get a more generalizable, more accurate, and more stable model.

First, the bootstrap method randomly selects training subsets with replacement from the entire dataset. Those not selected, taking about 1/3 of total samples, are "out of bag" (OOB) samples that are called out by the OOB method in ski-learn. Since they are never used in training, they readily test the model, and the method of reserving a subset for testing is no longer needed.

On each sample, random forest builds a decision tree using only a subset of all features. This increases the randomness within the model, which leads to more robust overall predictions. The individual models work in a similar recursive mechanism as detailed for Decision tree.

At last, after results are produced for each sample, the "majority votes" method is used to vote out the final prediction by aggregating all predictions from the decision trees. Each decision tree gives the probability that the instance is a popular track, and the "majority votes" method averages this probability to give final classification results. [20]

As a whole, random forest is more preferred compared to decision tree due to its larger diversity and randomness that prevents the problem of over-fitting and increases model accuracy by voting, which disregards instances of bad predictions. Very uncorrelated trees will grow from each sample and adds to the diversity of the forest. However, its interpretability is lower as affected by its complexity.

[3] Picture come from hvidberrrg.github.io/deep_learning/activation_functions/sigmoid_function_and_derivative.htm

4.1.5 Boosting Tree

Boosting Tree Explanation: Boosting tree is a classification model similar with random forest such that they both utilize Decision trees. However, boosting tree doesn't run trees in parallel and there is no voting process. Boosting tree is a sequential tree model that each learner in the sequence improves upon the previous training results by increasing the weight of errors and thereby trying to learn these errors within each iteration.

Each model is trained sequentially first, starting with a model having equal weights and then adding weights to errors from the previous model. After each classifier is trained, each is assigned a weight such that more accurate classifiers are assigned larger weights. The weight is calculated by $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$, in which α_t is the weights of the t^{th} classifier and ε_t is the weighed sum error for misclassified points, based on the classifier's error rate. As a result, the final classifier model is just a linear combination of all the individual classifiers' output times weight of each classifier.

$$(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \qquad (5)$$

in which $h_t(x)$ is the output of the t^{th} classifier (XGBoost Documentation).

Boosting Tree and PCA: With the boosting tree model, we used PCA, or Principle Component Analysis, to reduce features of the model and project it to a lower dimensional space. Sometimes, overfitting is caused by high dimensions, or in other words large number of features, and the fact that they are intercorrelated, which makes the weight of certain features diverge from their real contribution to the model. PCA prevents these interrelationships between features. PCA works by first identifying a hyperplane. The data are normalized, and PCA calculates the covariance matrix of the features involved. Eigenvalues and eigenvectors of the covariance matrix is then calculated, and the original data is multiplied with the eigen vectors which suggests the directions of the new axes. Then, the calculation results are plotted, which demonstrates how closely related the data are.

As a model using PCA is relatively unexplainable since the principle components don't have real world meaning as the original data do, the performance of the classification model using PCA components can be visualized by using TCSEVisualizer. The visualizer and PCA decrease dimensions of the data and show how well the test samples are predicted by using colors to denote popular or unpopular tracks, thereby showing how well the two groups are discriminated. The results give suggestions for whether predictability exists within music's popularity, a largely intangible measurement.

4.1.6 Neural networks

Neural networks are a model that is theoretically best used for highly nonlinear systems, but also easily overfits. As mentioned in Section 2, listeners nowadays tend to prefer music of higher diversity, so we hypothesize that there is hardly a linear trend between each feature and popularity since very popular music may have largely diverse audio features. In this case, neural network and some previous models that do not solely fit linear relationships may be a better for this investigation.

Neural network includes layers of "artificial neurons" that originated from the structure of biological neurons and were extended to methods of machine learning. In a Multi-layer Perceptron (MLP) model, one single neuron from one layer accepts input from other neurons from the previous layer, processes it, and passes to another on the next layer. This creates a feedforward artificial neural network.[4]

In each single neuron, input values are weighted, and the sum passes through an activation function. The hidden layer's neurons obtain the results from the previous layer, and passes on until meeting the end of all layers. Two other layers – input layer and output layer – don't do any calculation work but simply pass the data in and out of the system. By MLP, neural networks are able to predict linearly inseparable relationships.

In our model, we implemented the ReLU (Rectified Linear Unit) activation function in inner layers, which is currently the most used activation function throughout the world right now. ReLU creates a threshold such that all input values below it will output 0, which allows the network to converge must faster and creates a more practically and computationally efficient model. The final layer utilizes a Softmax function that assigns probabilities to each class by $\widehat{p_\iota} = softmax(z_i) = \frac{e^{z_i}}{\sum_j e^{z_i}}$ in which z_i represent that value of the i^{th} instance in the previous layer. In addition, in the input and hidden layers, a bias node is added to shift the activation function to the right or left to fit the data. As a whole, the output is calculated with the formula $a_{out} = g_k\big(b_k + \sum_j g_j\,\big(b_j + \sum_i a_i w_{ij}\big) w_{jk}\big)$, such that $\sum_i a_i w_{ij}$ is the summation at one node in i^{th} for its inputs times weight, w_{ij} is the weight connecting layer i and j, g_j is the activation function for this j^{th} layer, b is the bias term in each layer, g_k is the activation function for output layer, or the k^{th} layer. By such a computation fed-forward layer by layer, the final output is calculated.

The cost function of neural networks uses the sum of square errors, which is calculated at the end of the model. Similar as previous models, we hope to find the optimizing set of weights and bias in each layer that minimize $error = \sum(output - target)^2$.

To minimize the error, we use gradient descent to reduce error iteratively. New weights are obtained through moving the original weights along the multi-dimensional space with the formula ${w_{ij}}^{new} = {w_{ij}}^{old} - \eta\frac{\partial E}{\partial w_{ij}}$. A negative gradient increases weights and moves towards possible lower loss; a positive gradient decreases weights and also move towards the global minimum.

To find the optimizing set of weights, neural system uses backpropagation algorithm. This algorithm starts with the output layer k, and then to j, i, and so on, which back-propagates the error to previous layers and calculates the error at each output node including the hidden layers. The following calculates the partial derivative of error function E with respect to parameters between layers $l - 1$ and l.

$$\frac{\partial E}{\partial w_{l-1,l}} = a_{l-1}\delta_l \qquad (6)$$

In the equation, δ_l represents the back-propagated error term on node l, and a_{l-1} represents the output on node $l-1$, which is the previous node. Thereby, parameters are updated until the final output's loss converges. [21]

4.2 Hyper-parameters Tuning

For tuning hyper-parameters, we used GridSearchCV from ski-learn to evaluate different combinations of the values for these three hyper-parameters through cross validation and derive the best one that yields best evaluation results. GridSearchCV is a function that splits the training set into 5, and each set of hyper-parameters is used to train the set on four of the subsets and tested on the other. The average score of five fitting results for each combination of hyper-parameters are compared, and the best combination is chosen. The cross-validation method prevents over-fitting and bias due to smaller datasets since the model is trained on different subsets and averaged.

After using GridSearchCV and fitting the training sets, we call "best_params_" that gives the set of hyper-parameters that created best-performing models. [20]

In Logistic Regression, we tuned the hyper-parameter "C" which denotes the "inverse of regularization strength", or a measure of the inverse of constant m within the cost function. The larger "C" the stronger regularization. This parameter helps eliminate outlier weights and thereby increases generalizability of the model.

Decision Tree has a few hyper-parameters, including "max_depth", which is the maximum iterations that can occur or the maximum layers the decision tree has; "min_samples_split", or the minimum samples required to split an internal node; as well as "min_samples_split", the minimum number of samples required at one node after each split. An overly large maximum depth may make the model overfit the data and reduce its generalizability, while overly low minimum splits may lead to meaningless levels. [20]

In Random Forest, first, "max_depth" and "min_samples_leaf" are also used; "n_estimators" is the number of trees in the forest. In general, the larger "n_estinators" means that there are more trees within the forest, implying that the majority voting strategy would help eliminate more errors occurring in some of the individual tree models. [20]

In Boosting Tree, considering the complexity of boosting tree which takes a long running time, "max_depth" and "n_estimators" are two main hyper-parameters that are tuned carefully. Learning rate is set to 0.005 as a typical value. "subsample" is the proportion-wise sample size of the entire sample used during each iteration, which is typically set to 0.5. [20]

Different from other models, neural networks require manually creating the model parameters including learning rate, the number of nodes in each layer, the number of layers, as well as the activation function.

4.3 Evaluation metrics

The evaluation metrics used in this paper include the following: precision, recall, and F1 score. The former three evaluation metrics are derived from the confusion matrix, while F1 is a combined score of precision and recall. Given that our labels are highly imbalanced, the commonly used accuracy measure may yield misleading interpretations. Therefore, we decide not to include accuracy as the evaluate metrics of all models.

TABLE II. CONFUSION MATRIX

	Predicted Positive	Predicted Negative
Actual Positive	True positive (TP)	False negative (FN)
Actual Negative	False positive (FP)	True negative (TN)

4.3.1 Accuracy

$$\text{Accuracy} = \frac{TP+TN}{Total} \tag{7}$$

Accuracy denotes the proportion of number of correct predictions out of total predictions. This is one of the most straightforward measurements of a model's performance by assessing the general correctness of the model's predictive results. In fact, accuracy doesn't vary much across all models. Furthermore, we only want to focus on the tracks that are popular and according to what features the models assign the tracks as popular is what interests us. Therefore, other metrics are relied on more to effectively evaluate the models' prediction results.

4.3.2 Accuracy

$$\text{Precision} = \frac{TP}{TP+FP} \tag{8}$$

Precision measures the proportion of real positive instances that are successfully predicted out of all predictions of positive instances. In this paper, precision denotes the proportion of actually popular tracks within all tracks that are predicted as popular. Since what we want to focus on are the features that lead to high popularity of tracks, this precision score is very important as a higher precision score shows that the model has found the characteristics of popular tracks that allows it to correctly predict what tracks are popular.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{9}$$

Recall measures the proportion of predicted positive instances out of all positive instances, meaning the proportion of popular tracks that are successfully predicted as popular out of all popular tracks. A higher recall score in this paper means that the model is able to fetch a large number of tracks as popular out of the entire pool, meaning that the traits of tracks that the model finds that determine tracks to be popular are can be applied to many of those popular tracks. Therefore, the model can recognize a wider range of popular tracks, possibly of different types. This makes the model more comprehensive and therefore valuable.

4.3.3 F1 score

$$\text{F1 score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{10}$$

While there is usually a tradeoff between the precision and recall scores, the F1 score provides a combined score of the

[4] Basheer, Imad & Hajmeer, M.N. Artificial Neural Networks: Fundamentals, Computing, Design, and Application. Journal of microbiological methods. 43. 3-31. 10.1016/S0167-7012(00)00201-3. [22]

harmonic mean for the two, and gives a suggestion for what combination of precision and recall may create a best model. This F1 score allows us to choose from a range of models that have different levels of precision and recall that none of the models have absolute advantage over others in both of the evaluation scores. [20] This is also especially useful when both recall and precision are important in our model and models with different combinations of the two score are difficult to compare.

4.4 Model Interpretation Tools – SHAP

4.4.1 SHAP value

Once we have trained the best performing models, interpretability is an important problem since we hope to understand what the model suggests as the important features that make popularity predictable. This is also a problem that currently occurs throughout most models – more accurate models are usually more complex, which creates a crucial and unescapable tradeoff between interpretability and model performance.

We used SHAP to explain the models and how label values are affected by features. [20] SHAP is a widely use in machine learning model interpretation. SHAP value is a measure of the contribution of each feature on prediction, which gives an important reference for the model's real-life implications.

In this paper, SHAP calculates and visualizes the contribution of each audio feature on track popularity. Through classification models, we readily observe what features in what directions cause tracks to be popular. This gives economic implications for music production industries and hints on specific leans during the production process that may create music that are more likely to succeed.

In a simple version of explanation, SHAP "takes the base value for the dataset", in this case popularity of 0.38 (scaled) above which is classified as popular tracks. Then, SHAP "goes through the input data row-by-row and feature-by-feature varying its values to detect how it changes the base prediction holding all-else-equal for that row". Though such a process, SHAP is building a "mini explainer model for a single row-prediction pair to explain how this prediction was reached" (Evgeny 2019). This model measures the extent to which the prediction is altered through changing features, thereby gives a reference for each feature's importance on popularity, which explains the model with higher clarity.

4.4.2 SHAP model

After computing feature importance, we are able to produce a summary plot for the most significant features in this model. By visualizations and SHAP values, we can explain in what direction and magnitude each feature affects the prediction, thereby transforming numerical values into real-world meanings, specifically explaining what features in what direction and how strongly affect popularity of tracks. All of this information is summarized in the SHAP Summary Plot. In the plots, red and blue means high and low feature values respectively. The horizontal axis means negative or positive SHAP values from left to right, denoting negative or positive influence of certain feature values on the label value, or popularity.
We also created force plots built within SHAP in python. The force plots show how strongly each feature affects a certain predicted value. This implies each feature's importance in predicting the label that models real world instances, thereby showing whether there are and what features are deterministic in making tracks popular.

4.5 Classification Visualization

To visualize the effectiveness of using machine learning and models to classify tracks, we used TSNEVisualizer to effectively and simply do so. It is a useful tool that decreases the dimension of data from high dimensions to low or just 2 dimensions. This helps create a graph that clearly demonstrates how accurately tracks are classified based on PCA.

5 Results

5.1 Best Hyper-parameters for Models

Logistic regression.— $C = 0.05$, $max\_iter = 100$, $penalty = 'l2', tol = 0.0001$
Decision tree.— max_depth = 3 , min _samples_leaf = 3, min_samples_split = 5, criterion = 'gini'
Random forest.— max_depth = 40 , min _samples_leaf = 3, n_estimators = 70, criterion = 'gini'
Boosting tree with PCA.—eta = 0.002, max_depth = 60, objective = 'binary: logistic', subsample = 0.25 , early_stopping_rounds = 20 , verbose_eval = 1000 , n_components = 20
Neural networks.— Weight for class 0 = 0.67 , Weight for class 1 = 1.98 , kernel_initializer = 'he_normal', activation = 'relu', $layer\ 1 = (16, activation = "relu")$, $layer\ 2 = (8, activation = "relu")$, $layer\ 3 = (1, activation = "sigmoid")$

5.2 Best Hyper-parameters for Models

5.2.1 Metrics Summary

TABLE III. Evaluation Metrics Summary

	Accuracy	Precision	Recall	F1 Score
MLR (NE)	0.803	0.764	0.882	0.819
Logistic Regression	0.809	0.787	0.853	0.818
Decision Tree	0.793	0.737	0.918	0.818

Random Forest	0.827	0.819	0.843	0.831
Boosting Tree	0.829	0.823	0.840	0.832
Boosting Tree (PCA)	0.828	0.817	0.847	0.832
Neural Networks	0.831	0.828	0.840	0.834

Note: Here "MLR" is multiple linear regression; "NE" is by normal equation; "PCA" is with Principle Component Analysis

From the results, we can see that the evaluation metrics are all relatively high, all above 0.8, even the simplest baseline models linear regression, logistic regression, and decision tree. This has already been a huge improvement from the original dataset before having set up the quartile categorical variables for the artists' number of songs and artists' number of hit songs in an earlier dataset. Rather than having produced the two categorical variables, we used OneHotEncoder in python to set each artist as a categorical variable, which resulted in 138 features in total. However, the models built upon this processed dataset have F1 scores between 0.2 and 0.4, with the best about 0.5. This turned out that individual artists' names did not effectively indicate popularity, and rather than specific artists definitely produce tracks with high popularity, it is actually more possible that there is a general trend that artists with different number of tracks, as well as artists who have once had hit songs, might have larger relationship with music popularity, thereby showing a general trend rather than simply the preference for specific artists.

The performance of three baseline models – Multiple Linear Regression, Logistic Regression, and Decision tree – are relatively weaker, while the more advanced ones have generally and gradually increasing performance as measured through F1 and accuracy. Furthermore, it can be seen from the precision and recall scores, that baseline models have relatively less balanced scores, with a generally larger recall, while the better performing have more balanced and similar precision and recall scores.

For baseline models, there is a trend that recall is often higher than precision, while for others the two metrics are relatively balanced. Increased complexity allows the model to select popular tracks with a better accuracy, so that a larger proportion of tracks that are predicted positive are really positive, leading to more balanced metrics.

The accuracy of all models are gradually increasing, with the largest successfully predicting 83.1% of all tracks. Note that accuracy could be used as a metric since we balanced the positive and negative sets to the same number of samples.

Overall, the results from the evaluation metrics summary are relatively satisfactory although there are not large improvements from baseline models to others, possibly because the models are already relatively strong. These models can readily give some implications of listeners' music preferences just towards audio features. And show that these features can accurately predict more than 80% of whether songs are popular or not.

5.2.2 PCA Evaluation

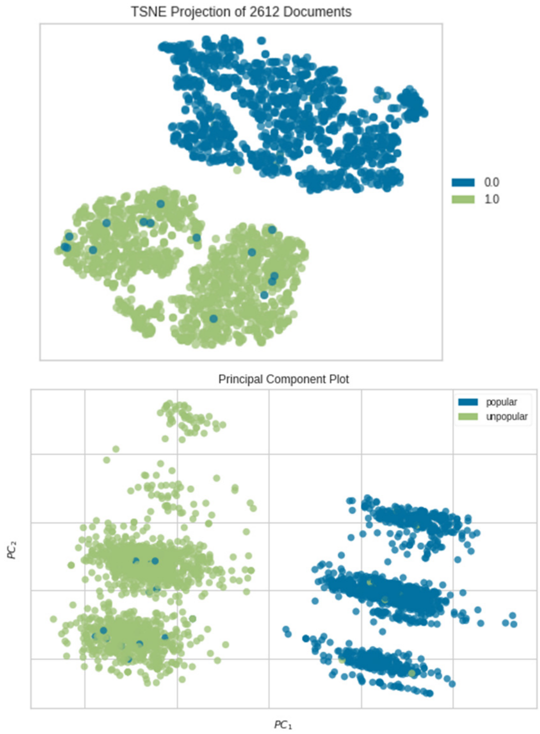

Figure 9. TSNEVisualizer AND Principle Component Plot Visualizations, Boosting Tree

To visualize how well song popularity can actually be predictable by audio features and artists, we utilized TSNEVisualizer to plot such graphs. The first plot is the result of using TSNEVisualizer to observe results of Boosting tree classifiers after having been decomposed into a two-dimensional space, and the second plot is a similar but considers only the first two principle components. After PCA has projected the 20-dimentioanl data after feature reduction of Boosting tree to the 2D space, from either of the two plots, we can see that when only considering the first two principle components, the clusters are already relatively clearly separated with rarely erroneously classified tracks. This further shows the effectiveness of Boosting tree model as well as PCA's usefulness. Songs are clearly and separated into two groups – popular and unpopular – with extremely clear discrimination between the two. This implies the problem is largely predictable only by including acoustic features and artists by simple visualization.

5.3 Interpret Features

5.3.1 SHAP Summary Plots and SHAP Values

https://doi.org/10.1051/e3sconf/202125303024

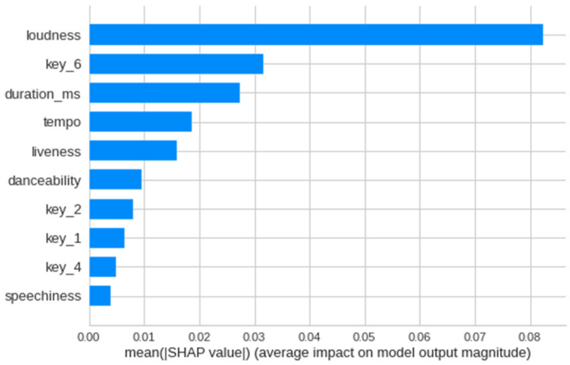

Figure 10. Boosting Tree Feature SHAP values

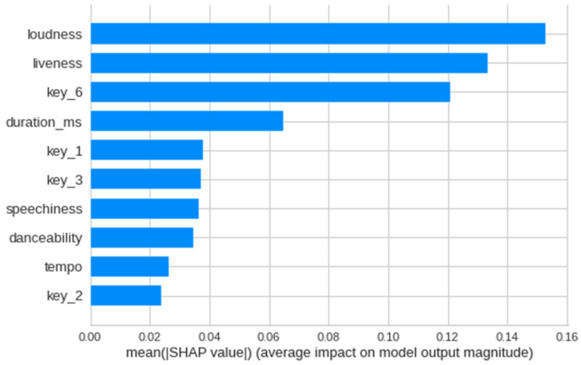

Figure 11. Neural Network Feature SHAP values

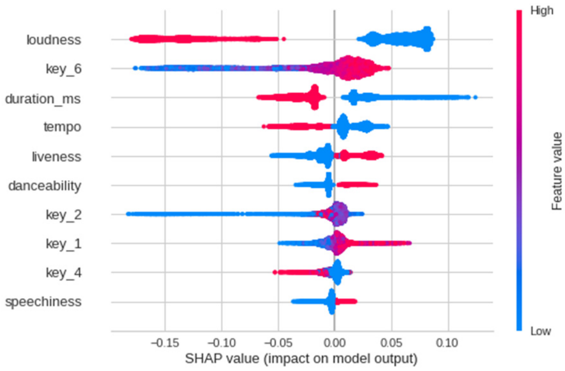

Figure 12. Boosting Tree SHAP Summary Plot

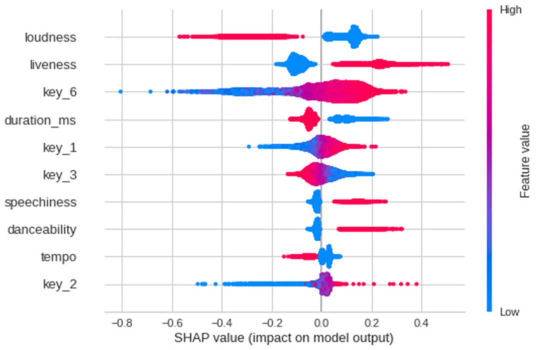

Figure 13. Neural Network SHAP Summary Plot

Within this section, we produced SHAP analysis for two models – Boosting Tree and Neural Network, two best performing models – which are looked in a combined way.

In the plot of absolute SHAP values, the larger SHAP value means this feature has a larger influence on whether music is popular. From the plot, we first notice that there are many "key_n" features that were dissected from the original key feature. Given that all tracks have one of these keys, when thinking of music theoretically, choosing a key might seem to be trivial within music production since to listeners without much knowledge of music theory, their listening experiences of a track are not largely affected by the key. However, according to our model, the key is largely affecting whether a piece of music is liked by listeners in general, with "key_6", or F♯, G♭, being the most influential with a noticeably high absolute SHAP value. Taking reference of the summary plot following, keys 1, 2, 6 have positive effects on popularity, while keys 3, 4 push track popularities towards the negative side.

Other than the key, the most important features in both models, loudness, appears to be very significant for tracks becoming popular, as it has a much higher SHAP value than all other features in the Boosting Tree model. This is an unexpected discovery from the models.

In the SHAP Summary plots, in which direction the features affect the popularity is clearly demonstrated. Although the feature importance of two models are slightly different, the common features denoted as important in both models affect popularity in common directions, denoting a correspondence in the two models' results and confirming each other.

One of the largest observations within the plots is the large importance of loudness on popularity, which was originally observed in the data exploration section to have the largest linear correlation with popularity by having the darkest color in the heat plot.

The summary plot shows that a higher loudness feature value leads to popularity of songs to be low, as denoted by the red region to the negative side of SHAP values. As demonstrated in data exploration, if a piece has occasional large volume or largely fluctuating volume, then its loudness would be relatively low as Spotify calculates it as an averaged value; if a song is of consistent high volume, then its loudness would be relatively higher. The results actually imply that songs either quieter or with more dynamic in volume and with more emotional changes makes music more attractive, compared to those that are constantly loud.

Furthermore, liveness, or the extent to which the music was played live, is a feature that positively affects popularity – higher liveness occurs more in popular songs – and could be seen as the second most influential feature in Neural Network model and also relatively important in Boosting Tree. This means that music with longer reverberation time, which sounds less monotonous and tedious.

Duration, in milliseconds, shows that the shorter duration of the track generally means better popularity, or that duration negatively affects whether tracks are popular. This implies that tracks that are very long generally loose listeners' patience and therefore have lower popularity. Though previous papers have stated that songs between three to five minutes are much easier to succeed and become hit songs, results show that between tracks being too long or too short, being too long is more unacceptable if the track's goal is to become popular.

Tempo has a relatively lower influence on both models, but is still in the top 10 important features and negatively influences popularity. Especially in Figure 13, the long red strip, on the row of tempo which is on the negative side of SHAP value, shows that high tempo generally causes songs to be less popular; the small blue cluster on the positive side of SHAP value implies that low tempo has little positive effect on popularity.

One significantly influencing feature is danceability, and its visualization of both SHAP summary plots are very similar: high danceability lead to largely high SHAP values, or high popularity, denoted by the red strip on the positive portion of SHAP values. Other than showing tracks with rhythms that can be used to dance are exciting, it also implies that possibly danceable tracks have wider applications and therefore more easily succeed. Also, it implies producing a track with its dance choreography or a separate dance mv can largely increase the possibility of the track to success.

At last, speechiness is also a feature that is positively affecting popularity. Tracks with larger speechiness, those with more words than music such as a speech audio or with more rap, have a larger possibility to succeed.

5.3.2 SHAP Force Plots

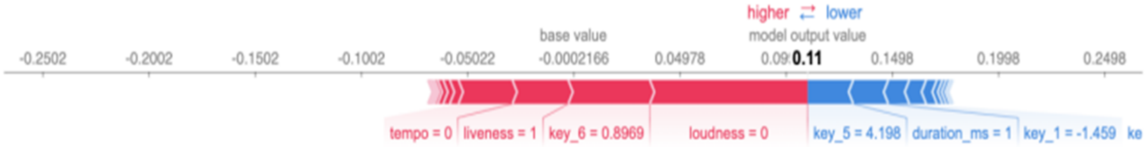

Figure 14. Sample Instance Force Plot of Boosting Tree

In the force plot for Boosting Tree, all of the features' values lead to the prediction value of 0.11, which is then converted into 0. In this plot, it is observed that loudness and key_6 push the predicted value larger, while those written in blue push the value smaller while only with small amounts. High liveness and low loudness, written in red, positively affect the predicted results, while high duration negatively affects the prediction. Thereby, most features that are on this plot correspond what is concluded to be important previously in the SHAP value graphs.

6 Discussions, Implications, and Conclusions

As shown in the results section, we can use the best performing models Boosting tree, Random Forest and Neural Networks to predict the popularity of music at about 0.82 precision and 0.84 recall, which together produce a F1 score as high as 0.83. Also, the best accuracy score is 0.831, meaning that more than four-fifth of all song tracks are predicted correctly, a large success in terms of as the test set that was not included during the training process, implying the models' high generalizability. These models perform relatively better than the baseline models – Multiple Linear Regression, Logistic Regression, and Decision tree. The small improvement from baseline models to more advanced ones may result from the fact that whether a song is popular is already well-shown by the numerical features included and categorical features set through the data exploration section, so that even the simplest models allow good performance.

One of the most important implications of the investigation is the high predictability of music popularity, solely by audio features and artists' profile. Given that music consumption has many uncertainties, including quality and value based on personal preferences, popularity may seem quite unpredictable. In real life, although some may believe that popular music seems to be more and more similar without much artistic value, there are also some music that are popular by other economic or social reasons such as the lyrics' meaning and advertisement strength. Therefore, our models imply high predictability of popular tracks solely form audio features and artists' profile, therefore is largely contributing as our focus actually is capable of giving certain implications toward the music production industry.

Using SHAP allows to generate a suggested list of features that may be used as a reference for music composition. Other than composing a repetitive melody that acts as an earworm within listeners' head, these audio features included provide suggestions to the most basic fundamentals of pieces of tracks such as the key it is composed in or its loudness, that can largely affect popularity as implied by the modeling results. Our model results imply that possibly slight changes in these fundamental and theoretical features may increase popularity of tracks to some extent. Meanwhile, our results also demonstrate how linear models have performed relatively weaker compared to models that can capture the non-linearity in the acoustic features of music, such as Boosting tree and Neural Networks.

Moreover, while the specific levels of popularity, or amount of recent streaming, is difficult to measure, the evaluation metrics show that the most songs can be found with about 83% accuracy according to audio features and artists' history profile. We have found noticeable similarities between the tracks that have become popular by simply accounting for the acoustic features, because the music industry might have intentionally produces hit tracks that are so hard to ignore, a largely profitable method in production of music, which consumers seem largely unpredictable.

In our model features interpretation, using SHAP, there are several features that stand out and demonstrate importance in their contribution to tracks' popularity. First, audio features liveness, danceability and speechiness positively affect whether tracks are popular; in other words, the larger value in these features means higher possibility that the track passes the threshold for popular tracks. Other important features in Boosting Tree and Neural Network models are loudness, duration and tempo that negatively affect popularity; lower values in these features push tracks to pass the threshold of tracks being popular. Listing these features in decreasing importance on the models, we have loudness the most influential, then liveness, duration, tempo, danceability, and then speechiness. The key of a track is also important for a successful track; specifically, keys in 1, 2, 6 have positive effects on popularity, while keys 3, 4, have negative effects. Surprisingly, rather than artists having largest effects on music popularity, although they might play a relatively important role, the most fundamental audio features, even keys that nonprofessional listeners rarely notice different in tracks, are having large effects on popularity.

Given many implications of the modeling results in terms of features' importance and what they tell about music in general, we can see several meaningful conclusions and utilizations. The most important implications of such a music production implication can hold different meanings to different groups of people. To music producers, having a reference of important musical components that can possibly produce popular tracks may allow producers to compose tracks based on certain guidelines for the fundamental elements of a track, such as its dynamic in volume, which is related to loudness, its key, and whether beats are suitable for dance or producing dance music videos, etc. This type of music production with the intention of gaining popularity by acting in accordance to certain guidelines allows music production companies to dissect what consumers really like, and thereby make the greatest profit out of this type of consumer analysis using machine learning.

However, this usage of such a model in music industries might draw laybacks. In perspective of music industry as a whole, analyzing the interest of consumers has always been an important process in making executive decisions about its products, which is tracks and albums. When this practice goes to the extreme and many music producers compose based upon certain guidelines and tips that are discovered could help produce commercially successful track, the intentions of music production may turn extremely monotonous. Music, which origins as a form of art, might lose part of its originality and traditional intention of expressing thoughts and feelings. To those composers with such passions and with noncommercial intentions, suggesting them to conform to certain formulae might harm incentives of producing music they like. Therefore, the results of this type of models should not be directly used as instructions of music composition, but should simply act as reference and hints for music producers within the industry, hoping to make the most profit. Note that one conclusion from our model is that music with diversity is relatively more preferred by consumers. This fact together with machine learning models' implications of specific music production elements may improve upon the profit conditions of music producers, but should be used and considered with care.

In conclusion, according to our models' results, other than the specific features that determine popularity, the most important conclusion is – audio features solely can make popular songs very predictable. This can be treated as an important discovery as machine learning can be used as an effective method in predicting consumer preference in such a digital market with goods which quality is largely subjective and intangible.

As human taste is an always changing and music is a unique and intangible "product", our model results only serve as a reference, and there are several limitations that may be improved upon in our model. First, many of the audio features, other than valence and key, are not specifically looked at due to relatively lower SHAP values. Although they might not be significant in determining popularity, the direction of their effects on popularity may also be interesting facts to look at. Furthermore, other than audio features, lyrics as another internal feature of music may also take a part in determining popularity. Keywords within lyrics as well as length or language of lyrics can also be treated as a feature and be modeled on. This work in the future may give more information for the preference of music consumers and provide specific types of lyrics that increase popularity of tracks. The additional features of lyrics may possibly add upon the current models and increase their performances.

References

1. Connolly, Marie & Krueger, Alan B., 2006. "Rockonomics: The Economics of Popular Music," Handbook of the Economics of Art and Culture, in: V.A. Ginsburgh & D. Throsby (ed.), Handbook of the Economics of Art and Culture, edition 1, volume 1, chapter 20, pages 667-719, Elsevier.
2. "Regression Analysis and Least Squares." VRU, 29 Mar. 2018.
3. Passman (2019), All You Need to Know About the Music Business: 10th Edition, Simon & Schuster, US.
4. Aguiar, L. & Joel Waldfogel 2018. Platforms, Promotion, and Product Discovery: Evidence from Spotify Playlists; JRC Digital Economy Working Paper 2018-04; JRC Technical Reports, JRC112023.
5. "Earworm." Merriam-Webster.com Dictionary, Merriam-Webster.
6. Seabrooks (2016). The Song Machine: Inside the Hit Factory. W. W. Norton & Company.
7. Jakubowski, Kelly & Finkel, Sebastian & Stewart, Lauren & Müllensiefen, Daniel. (2016). Dissecting an Earworm: Melodic Features and Song Popularity Predict Involuntary Musical Imagery.. Psychology of Aesthetics, Creativity, and the Arts. 11. 10.1037/aca0000090.

8. Trendjackers Team. “How Social Media Has Affected the Music Industry.” Trendjackers, 27 Jan. 2017.
9. Nam, Juhan & Choi, Keunwoo & Lee, Jongpil & Chou, Szu-Yu & Yang, yi-hsuan. (2019). Deep Learning for Audio-Based Music Classification and Tagging: Teaching Computers to Distinguish Rock from Bach. IEEE Signal Processing Magazine. 36. 41-51. 10.1109/MSP.2018.2874383.
10. Lundberg, Scott & Lee, Su-In. (2017). A Unified Approach to Interpreting Model Predictions.
11. Krueger, Alan B. “Rockonomics by Alan B. Krueger: 9781524763718: PenguinRandomHouse.com: Books.” PenguinRandomhouse.com, Crown, 2019.
12. Aguiar, L. & Martens, Bertin. 2016. Digital music consumption on the Internet: Evidence from clickstream data. 34. 27-43. 10.1016/j.infoecopol.2016.01.003.
13. Gauvin, H.L. (2018). Drawing listener attention in popular music: Testing five musical features arising from the theory of attention economy, Musicae Scientiae, 22(3): 291-304.
14. Myra Interiano, Kamyar Kazemi, Lijia Wang, Jienian Yang, Zhaoxia Yu and Natalia L. Komarova, Musical trends and predictability of success in contemporary songs in and out of the top charts, Royal Society Open Science, 5(5):171274.
15. Salganik, M & Dodds, Peter & Watts, Duncan. (2006). Experimental Study of Inequality and Unpredicatbility in an Artificial Cutlural Market. Science. 311. 854-856.
16. McKinney, Kelsey. “A Hit Song Is Usually 3 to 5 Minutes Long. Here's Why.” Vox, Vox, 18 Aug. 2014.
17. Askin, Noah & Mauskapf, Michael. 2017. What Makes Popular Culture Popular?: Product Features and Optimal Differentiation in Music. American Sociological Review. 82. 10.1177/0003122417728662.
18. Herremans, Dorien & Martens, David & Sörensen, Kenneth. 2014. Dance Hit Song Prediction, Journal of Musical Research, 43(3):291-302.
19. Araujo, Carlos & Cristo, Marco & Giusti, Rafael. 2019. Predicting Music Popularity Using Music Charts. 859-864. 10.1109/ICMLA.2019.00149.
20. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
21. Singh, Devang. “Neural Network In Python: Introduction, Structure and Trading Strategies.” QuantInsti, QuantInsti, 27 Apr. 2020.
22. Basheer, Imad & Hajmeer, M.N.. 2001. Artificial Neural Networks: Fundamentals, Computing, Design, and Application. Journal of microbiological methods. 43. 3-31. 10.1016/S0167-7012(00)00201-3.
23. Dewan, S., Ramaprasad, J., 2012. Music blogging, online sampling, and the long tail. Inf. Syst. Res. 23 (3-part-2), 1056–1067. doi:10.1287/isre.1110. 0405.
24. Evgeny Pogorelov. “Explaining Multi-Class XGBoost Models with SHAP.” Evgeny Pogorelov, 13 May 2019.
25. “Experience Good.” Market, market.subwiki.org/wiki/Experience_good.
26. “The Long Tail Theory, Debunked: We Stick With What We Know.” Mack Institute for Innovation Management, 14 Nov. 2019.
27. Nagpal, Anuja. “Principal Component Analysis-Intro.” Medium, Towards Data Science, 22 Nov. 2017.
28. “Sklearn.decomposition.PCA.” Scikit.
29. Tianqi Chen, Carlos Guestrin, “XGBoost: A Scalable Tree Boosting System”.
30. “XGBoost Documentation.” XGBoost Documentation - Xgboost 1.3.0-SNAPSHOT Documentation, xgboost.readthedocs.io/en/latest/index.html.