Can You Predict a Song's Commercial Success?

ORIE 4741 Final Project

By: Darren Ting, Joshua Kwasi Agyei-Gyamfi, and Mario Medina

December 13th, 2020

1 Abstract

Can we use supervised machine learning to predict **commercial** success of songs using auditory features? We explore this question with the motivation of identifying popular factors to help musicians better understand what makes music popular. To answer this question, we utilized multiple datasets containing popular song lists as well as lists containing songs spanning two decades. We used techniques such as upsampling and grid search to fine tune our models to classify binary success of songs. For our best performing model that uses the XGBoost framework, we report an accuracy of 88.8% with a recall score of 53.3%. Through our thorough analysis, we find that, while we are able to achieve promising results, there are countless other social and cultural factors not accounted for in auditory features that motivate consumers to listen to the music they listen to. Given these extraneous factors and other limitations, we recommend further analysis into the cultural effect of music popularity. In this report, we explain the methodology behind our predictive model, highlight important trends and results, and probe what possible implications these results could have.

2 Introduction

Often times, when musicians begin the songwriting process, it would be common for them to wonder how well their new song could potentially do. They may ask themselves, "Will it be played on the radio?" or "Will the newer generation enjoy it?". This mindset is especially prevalent in executives at top record labels, who sign contracts with artists and allocate millions of dollars to who they think will be the most profitable for them. Being able to predict the success of a song can potentially estimate its ability to bring in revenue as well as go mainstream and make an influence in today's society. By being able to accurately predict the success of a song, we believe that current musical trends in mainstream culture can be learned by both musicians and record labels to help tailor their music accordingly. For this project, we explore this further by asking, "Can you predict a song's commercial success based on the song's auditory characteristics?".

To allow us to answer this question, we utilize several machine learning techniques to develop a model that classifies a song as commercially successful or not. As input variables, we use audio feature data pulled from the API of the popular music streaming service, Spotify [1]. This data was used in conjunction with Billboard Hot 100 Chart data [2], RIAA Certified singles data [2] and Spotify weekly top 200 streams [2]. For the purposes of this project, **we define commercial success as appearance on any of these three charts**. Although we recognize that many other metrics can also be indicators of commercial success, this is the framework we decided to proceed with. The Billboard Hot 100 list is a record chart that tracks single song sales and streams. The RIAA certified singles data shows which singles sold at least 500,000 copies or 750,000,000 streams. Finally, the Spotify weekly top 200 streams is a weekly recorded chart that simply lists the top 200 most streamed songs globally that week. We use these datasets to create the binary response variable for our model. We believe that these datasets will allow us to successfully answer our question and give us more insight on what attributes allow a song to.

3 Data

The Spotify [3] and Billboard datasets, which we used, were obtained from Kaggle resources and created by Daniel DeFoe and Yamac Eren Ay, respectively. The music data we used spanned two decades, from 1999 to 2019. We used a Python script to recover the data (4 files) from the data source. From here, we originally used song name and artist pair to mark songs on the dataset containing the auditory features as successful; however, there were many issues with discrepancies in format between the multiple datasets. As a result, we pivoted to a new method: utilizing Spotify's API and Search function. Spotify attaches a unique ID to each track, which was given in the Spotify auditory features dataset but not in the chart datasets that we use to label commercial success. To correctly match songs between our chart data and auditory feature data, we use Spotify's API to search for each song and artist pair in the chart data, obtain the corresponding track ID, match that ID to the ID in the auditory features dataset, and finally add a label '1' to indicate a successful song. An in-depth description of the auditory features we used are in the table in **Figure 1.**

3.1 Data Description

Name	Range	Description
Acousticness	(Ranges from 0 to 1)	The relative metric of the track being acoustic
danceability	(Ranges from 0 to 1)	The relative measurement of the track being danceable
Energy	(Ranges from 0 to 1)	The energy of the track
Duration-MS	(Integer ranging from 200k to 300k)	The length of the track in milliseconds (ms)
Instrumentalness	(Ranges from 0 to 1)	The relative ratio of the track being instrumental
Valence	(Ranges from 0 to 1)	The positiveness of the track
Tempo	(Float ranging from 50 to 150)	The tempo of the track in Beat Per Minute (BPM)
Liveness	(Ranges from 0 to 1)	The relative duration of the track sounding as a live performance
Loudness	(Float ranging from -60 to 0)	Overall loudness in decibels (dB). The values are averaged across the entire track.
Speechiness	(Ranges from 0 to 1)	The relative length of the track containing any kind of human voice.
Year	(Ranges from 1999 to 2019)	Year of release.

Name	Range	Description
mode	(0 = Minor, 1 = Major)	Major or Minor key.
explicit	(0 = No explicit content, 1 = Explicit content)	The binary value whether the track contains explicit content or not
Is Successful	(0 = not successful , 1 = is successful)	Whether or not the song is commercially successful based on our criteria.

Name	Range	Description
Key	(All keys on octave encoded as values ranging from 0 to 11)	C as 0, C# as 1, etc.
Popularity	(0 - 100)	A ranking of how often a song is played in a recent span of time

Figure 1: *Spotify Auditory Features:* The top table shows all of the numeric variables, the middle table shows boolean variables and the last table shows ordinal variables.

3.2 Exploratory Data Analysis

After the Spotify data were merged with the Billboard, RIAA and Spotify streaming data, we were then able to then perform some exploratory data analysis. In total, our dataset contains 41,900 song entries, 6,343 of which have been classified as commercially successful songs, or 15.1%. As expected, there are more unsuccessful songs than successful songs, which creates a class imbalance may introduce bias to the models that we test. We later show methods to try and rectify this issue. First, we compare the differences between the mean song traits of successful and unsuccessful songs. Here, we define the difference between a mean trait, $\bar{x}$, as: $\frac{\bar{x}_{suc}-\bar{x}_{unsuc}}{\bar{x}_{unsuc}}$. From **Figure 2**, we notice that seven of the 14 mean song traits in successful songs are notably different (>10%) from the mean song traits in unsuccessful songs, indicating that there exists some difference between successful and unsuccessful songs that can be leveraged by a machine learning algorithm. To show the correlation between different song trait variables and also the correlation between song traits and the success of a song, we use a heat map as shown in **Figure 3**.

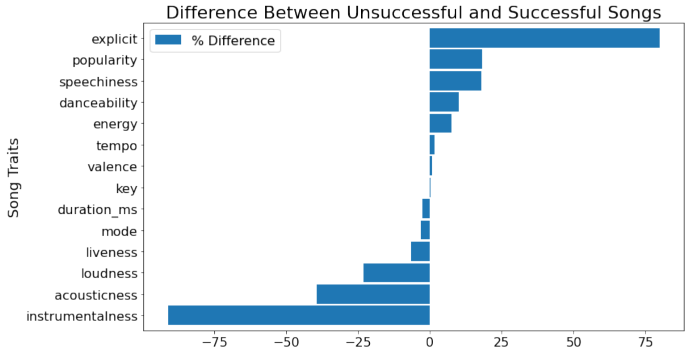

Figure 2: *Mean Differences:* This figure highlights which traits of successful songs are drastically different from the traits of unsuccessful songs. Successful songs are on average more explicit than unsuccessful songs and are much less instrumental.

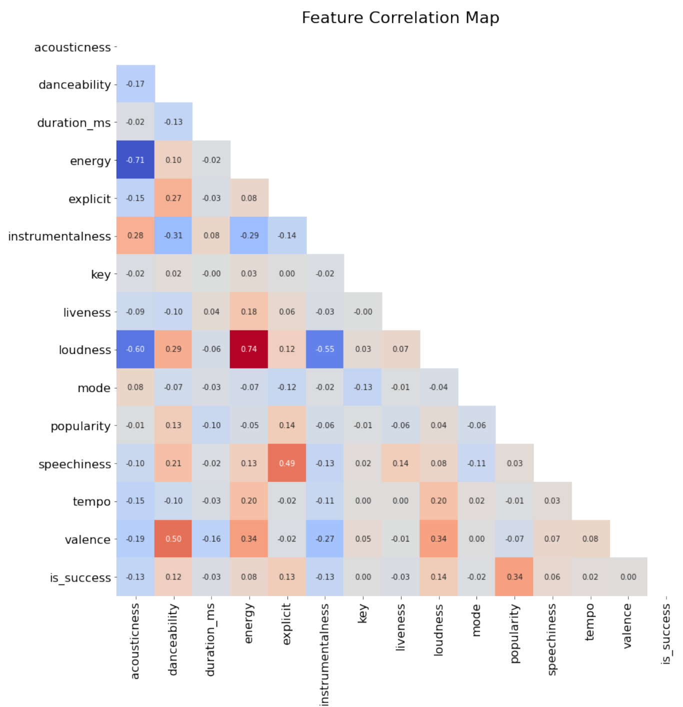

Figure 3: *Feature Correlation Heat Map:* Some variable pairs appear to be significantly correlated such as Energy being highly positively correlated with Loudness, Loudness being negatively correlated with Acousticness and Instrumentalness, and Energy being negatively correlated with Acousticness. For the correlation of song traits with the success of a song, Popularity showed the highest positive correlation and Instrumentalness showed the highest negative correlation, albeit both were still considered weak (0.34 and -0.13, respectively).

Next, we visualize the distributions of our song traits are shown in the plots in Figure 4. To obtain these, song trait values that weren't already given between 0 and 1 by the Spotify API were normalized by dividing by the maximum trait value. From these figures, we notice outliers in our data, however we choose not omit them as to increase the robustness of our model. These outliers will also influence which loss function we choose, as we must choose one that is insensitive to outliers.

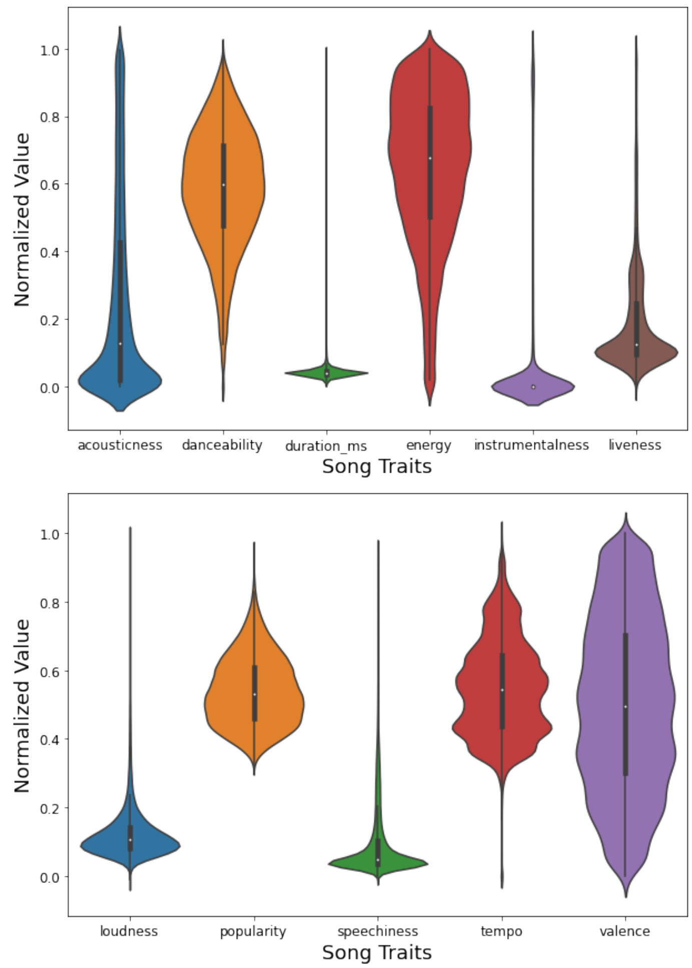

Figure 4: *Trait Distribution Violin Plots:* These figures show notable outliers in the Duration (ms), Liveness, Instrumentalness, Loudness, and Speechiness, and Tempo features, indicated by the thin, sharp jumps in the violin plot. Other traits such as Acousticness, Danceability, Energy, Popularity, and Valence are pretty well distributed.

3.3 Feature Engineering

To extract more information from our data for use in our model, we implemented two feature engineering techniques. One technique we used which showed improved results was word embeddings to find a way to incorporate artist name to our data. We used the open-source word embeddings model, GloVe [4] to numerically represent the words in the artist name. More specifically, they are word embeddings of N dimensions pre-trained on two billion tweets, where N can be chosen to be 25, 50, 100, or 200. The size of data used to train the embeddings ensures that the word embeddings we use should be accurate. For each song data point, we took the artist name, and summed up their word embeddings. For example, for representing the "Foo Fighters", we added the embedding vector element-wise for "Foo" and "Fighters" and appended it to the song row.

A second feature engineering tactic we attempted was using a "similarity" metric for each song. For this, we took the top ten songs that have been able to remain the longest at No. 1 on the Billoard Hot 100 [5] and used the Spotify

API to record their audio features as a vector. Then, the audio features of each song in our dataset were compared to each of the ten songs using cosine similarity, adding ten new features. This cosine similarity measure was inspired by its use in image processing and bioinformatics. This metric is defined for vectors u and v as:

$$Similarity = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||}$$

Unfortunately, this feature did not positively effect our model and was not used for later trials, as it lost accuracy compared to our baseline while retaining the same recall score. Recall is defined as the ratio of true positives and real positives, so in our case, that is the number of successful songs accurately classified over the number of successful songs. This metric will be commonly used in the following sections.

4 Analysis

4.1 Models

To tackle the question of whether or not it is possible to classify songs as commercially successful, we focused on using two models: XGBoost and the SVM. Our preliminary analysis showed that other models such as logistic regression and single decision trees did not perform as well as these two models.

4.1.1 XGBoost

One model we chose to use is the XGBoost model [6]. XGBoost is an optimized distributed library for Gradient Boosted decision tree algorithms. The XGBoost model uses a decision tree ensemble, and combines the outputs of the many trees to produce a final output. The XGBoost tree optimizes on the objection function

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y_i}) + \sum_{k}^{K} \Omega(f_k)$$

, where l is the loss function, y is the label, $haty$ is the prediction, f_i is the function, and Ω is the regularization. It utilizes additive training in that it fixes what the current number of trees has learned so far in the process, and then adds another tree to the ensemble. We minimize the objective function and the leaf weights so that the sum across all trees of the model are properly adjusted to minimize loss.

4.1.2 SVM

Another model we used was the Support Vector Machine, or SVM. The SVM, like the single perceptron model, creates a decision boundary between the two possible outcomes of a variable; however, what differentiates it from the perceptron is that it finds the optimal boundary that maximizes the distance between the closest points to the boundary and the boundary itself. The support vectors are the nearest data points to the boundary. Because data are not always linear, neither is the SVM boundary. When dealing with inseparable data given the current dimensions, it will project the data to a higher dimension to force it to be separable, unlike the perceptron which will not converge if data is not linearly separable. **Figure 5** shows how this is done in two dimensions.

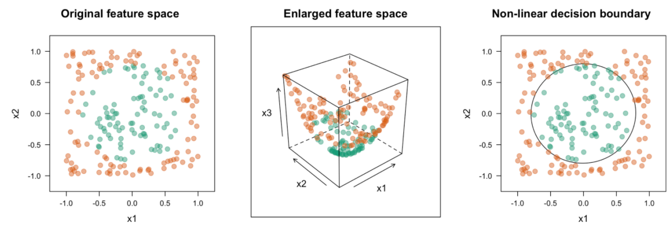

Figure 5: *How the SVM Works:* The left graph shows two-dimensional data which we would like separated into its green and orange classes. The SVM projects this data into a third dimension as shown in the middle graph, which is seen to be easily separable by a 2-D hyperplane. The right graph shows this plane as a boundary line in two dimensions. [7]

4.2 Model Tuning

Some of the methods we used includes K-Fold Cross Validation, grid search, regularization, and up/downsampling.

For our models, we used L2 regularization to reduce overfitting on our training set. Initially we trained our models without regularization and found them to either predict all songs as unsuccessful (85% accuracy, 0% recall), or all songs as successful (15% accuracy, 100% recall). Regularization reduces overfitting by penalizing large weight coefficients in the model, reducing the complexity. In doing so, computational time in training the model is reduced as well as prediction based on one feature alone. L2 Regularization differs from that of L1 Regularization in that L1 can reduce certain feature weights to zero while L2 ensures that the weights have smaller values.

K-Fold Cross validation (CV) is a technique used during model training in which the data is divided into k folds. One fold is kept aside as a testing set. The other $k-1$ folds are used to train the model individually and model performance metrics are averaged across k attempts, with the k'th attempt using all the data. This technique is implemented to avoid overfitting on the training set as well.

Grid Search refers to the hyperparameter optimization technique where we pass a grid of parameters and try running all different combinations of parameter values in the grid. We then choose the setting with the best result based on a metric of our choice (which in our case was accuracy).

Because the amount of unsuccessful songs in our data outnumbered the successful songs by about seven to one, our model tended to favor classifying most songs as unsuccessful. As a result, our model showed overfitting and predicted with a misleadingly high accuracy but a low recall. To circumvent this problem, we implemented upsampling and downsampling techniques. To begin, we split the data into training and testing sets. The training data is further split into majority (unsuccessful) and minority (successful) sets. In upsampling, the minority set is randomly sampled

n times with replacement, where n is the number of data in the majority set, giving us an equally balanced dataset of size $2n$. Likewise, downsampling randomly samples the majority class m times, where m is the number of data in the minority set. As shown in Figure 6, both downsampling and upsampling sacrifice accuracy for a higher recall, however this trend is much more noticeable for downsampling, where recall is increased by a factor of 6, which is close to the size of our original class imbalance. This can be explained by the fact that less data are repeated with downsampling, as the majority class has more data to randomly choose from. This allows the model to more accurately classify a successful song at the cost of overall accuracy. While upsampling still shows some signs of overfitting, it still shows less overfitting than if the technique wasn't used, shown by the negligible decrease in accuracy but 5% increase in recall.

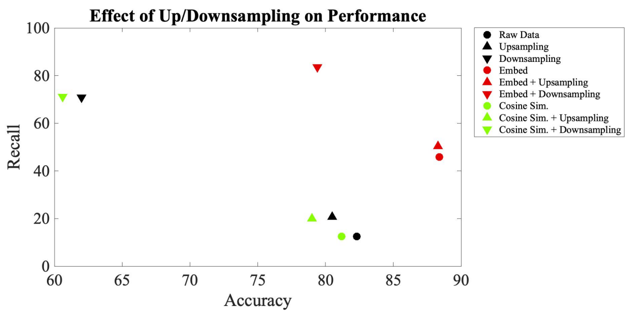

Figure 6: *Up/Downsampling for Imbalanced Classes:* We were able to tune the performance of various models using upsampling and downsampling on three datasets- our baseline, with word embedding features added, and then with the cosine similarity features added. The baseline and cosine similarity trials that used downsampling saw an increase in the recall score of nearly 70% at the expense of about 20% accuracy. The trials which used word embedding features changed less drastically than the other two, but still followed the same trend.

4.3 Results

After data analysis, feature engineering, fine-tuning of parameters, and trial-and-error (Figure 7), we decided on a final dataset and a best model. This final dataset consists of 41,900 songs between 1999 and 2019, which included 6,343 songs deemed commercially successful. In addition to the 15 auditory features mentioned in Data Description, we now include artist name as a 200-dimension word embedding. Our best model used the XGBoost decision-tree ensemble algorithm trained over 100 epochs, with logistic loss as the loss function (**Figures 7 and 8**). Our best performing hyperparameters found using k-fold cross validation and grid search are–Max Tree Depth: 100, Learning Rate: 0.3, L1 Regularization (α): 0.0001, and L2 Regularization (λ): 0.01. This model achieved an accuracy of 88.8% with a recall score of 53.6%. The feature importances for this model, along with others are explored in **Figures 10, 11 and 12**.
To further examine musical trends, we then divided the dataset into two by decade, so that we have a set of 22,000 with 2,548 commercially successful songs representing the 2000s decade, and a set of 18,000 with 3,472 commercially successful songs to represent the 2010s decade. This was done to potentially reduce the number of trends that had changed throughout the years, so that the classifier can perhaps perform better by learning the preferences of people during those decades (**Figure 9**).

Data	Model	Accuracy
Traits	SVM	83.9%, Recall: 0.001%
Traits	XGBoost	82.3%, Recall: 12.5%
Traits, Upsampling	SVM	84.8%, Recall: 0%
Traits, Upsampling	XGBoost	80.5%, Recall: 20.7%
Traits, Downsampling	XGBoost	62%, Recall: 70.9%
Traits, Artist Embed (25), Upsampling	XGBoost	88.3%, Recall: 50.3%
Traits, Artist Embed (50)	XGBoost	88.2%, Recall: 52.6%
Traits, Artist Embed (100)	XGBoost	88.4%, Recall: 52.9%
Traits, Artist Embed (200)	XGBoost	88.8%, Recall: 53.6%
Traits, Artist + Song Name Embed (50)	XGBoost	88.7%
Traits, Artist + Song Name Embed (25)	XGBoost	88.6%
Traits, Artist + Song Name Embed (25)	SVM	84.7%
Traits, Artist + Song Name Embed (25), Upsampling	XGBoost	88.7%
Traits, Artist + Song Name Embed (25), Downsampling	XGBoost	79.4%, Recall: 83.6%
Traits, Cosine Similarity	XGBoost	80%, Recall: 20%
2000s Songs, Traits, Artist Embed (25), Upsampling	XGBoost	90.6%, Recall: 49.4%
2010s Songs, Traits, Artist Embed (25), Upsampling	XGBoost	84.5%, Recall: 51.3%

Figure 7: *Feature Ablation Results:* We sought to try our models on a number of different features. We originally wanted to focus on auditory features (which we treat as our baseline), but ultimately decided to incorporate features such as popularity and artist name to see its effects on classifier accuracy. We also include the results for upsampling and downsampling. We use XGBoost primarily because while its baseline performed marginally worse than the SVM, it trades off with a much higher recall. We found that overall, downsampling vastly improves the recall metric at the cost of accuracy. In addition, incorporating artist name through word embeddings overall improved the model, but at a diminishing return and with a greater computational cost. Surprisingly, we found that adding song names in the form of word embeddings actually did not improve our model as much as artist name embeddings did. We believe this is due to the fact that many popular songs have simple titles, so that they are easier to remember. As a result, they may be more generic, and thus have less meaning behind it. Our cosine similarity feature did not perform well compared to others, losing accuracy while remaining at a constant recall. Of all the features and models used, we found that using 200 dimension artist word embeddings combined with upsampling in addition to the original auditory features allowed us to predict the best.

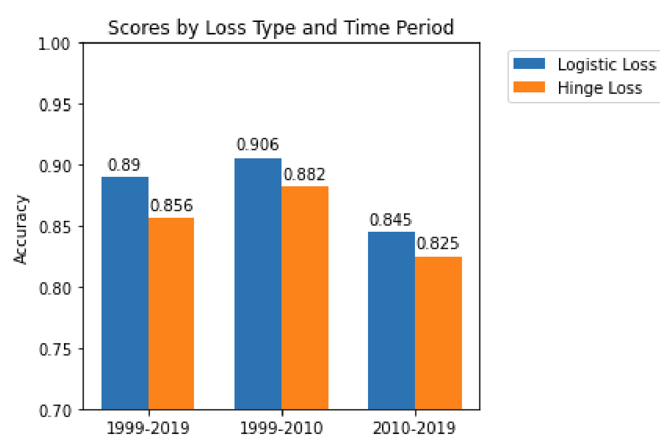

Figure 8: *Loss Functions:* This shows our best XGBoost model with both Binary Logistic Loss and binary Hinge loss. We recorded the accuracy across different decades as well as the overall two decade dataset. This decrease in accuracy could perhaps be due to the logistic loss retaining the probabilities of different classifications, while the hinge loss may be oversimplifying it.

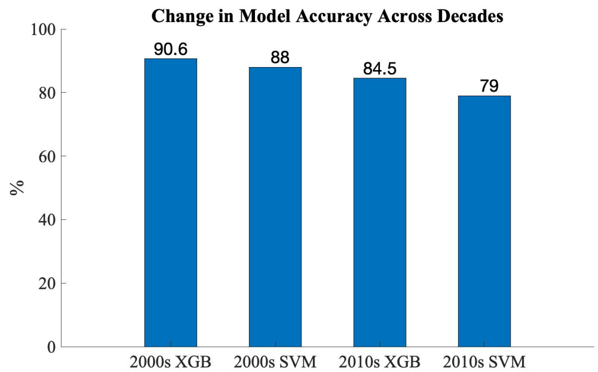

Figure 9: *Time's Effect on Model Accuracy:* Interestingly, we found that the performance of our models change when trained on data from different time periods. When we split the data into two decades, 2000s and 2010s, the model for the earlier decade performed better than the recent decade. This is because the word embeddings for artist names we used as features are not able to predict as well for the 2010s. In both decades, the XGBoost model performed better than the SVM model. It is worth noting that the XGBoost model for the 2000s data performed better in accuracy than our best model across all time periods by about 2%.

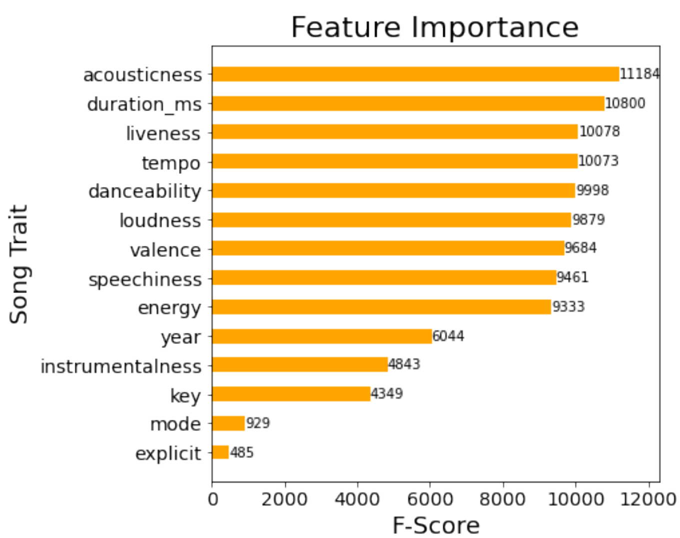

Figure 10: *Feature Importance (No Popularity, No Embeddings):* Here, we show the feature importance of the different auditory variables with respect to our best XGBoost Model. F-Score shows how often the variable is split on in the tree ensemble. Based on the graph, acousticness, duration and liveness are shown to be important to the model. As shown from our exploratory data analysis plots, live songs were generally not considered to be as popular as ones recorded in studio, and that acoustic songs on average tend not to show up on these different charts. The song duration may be explained as many songs that exceed a certain length may lose listener interest, and would decrease the number of streams, while a shorter song can increase its number of streams in the same amount of time.

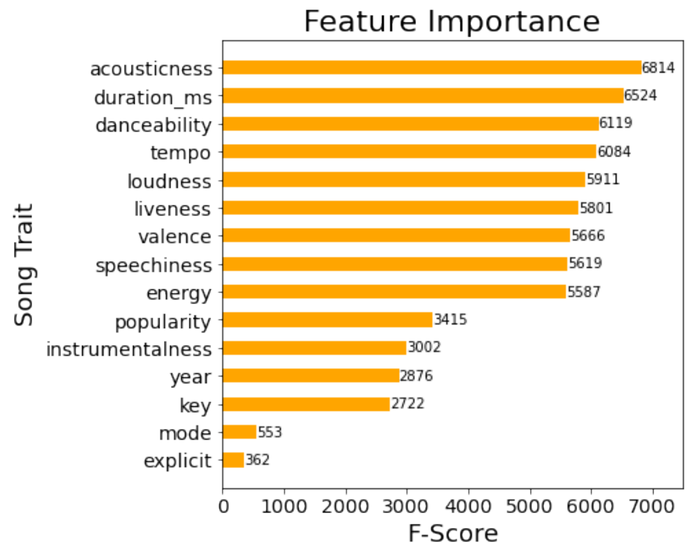

Figure 11: *Feature Importance (No Embeddings):* In our model where we add popularity, it turns out the variable does not seem to be taken into account as much. As it appears that other variables from the previous graph still weigh more.

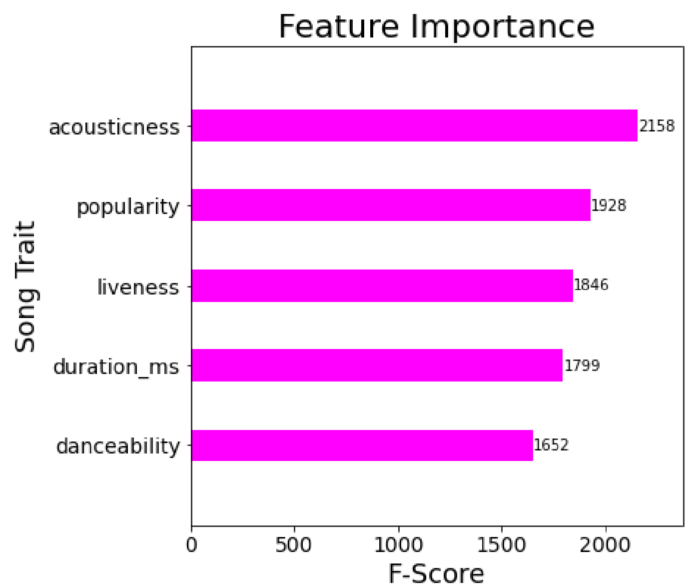

Figure 12: *Feature Importance (Best Model):* For our best model, we picked the five most important features that the XGBoost uses to split its decisions on. Here, we found that popularity was among one the most important features, which makes sense as Spotify quantifies popularity on a scale of 1 to 100 using the amount of streams in a given time frame. If the song is considered more popular, it probably is going to be more commercially successful. When we added artist name, popularity becomes one of the most important feature compared to the previous graph, where it was very low. In addition, the F-Scores are overall lower because of the number of feature dimensions we added to account for the artist name means that there are more features to split on, thus distributing the total number of splits to other variables.

5 Discussion

5.1 Findings

We originally wanted to focus on auditory features alone. We eventually incorporated the artist name through the word embeddings. By doing so, we represent the effect artist reputation has on the song's chances of making it on one of these well known charts which subsequently improved our model performances by 5%.

When we split up the datasets into decades, we found that models performed better in the 2000s than in the 2010s. We believe that the internet is a factor in this. Before the rise of social media, artists' music was more separated from their personal image and reputation. Thus, it was very difficult to make a name for yourself and artists with well-established reputations were able to keep finding success in their music; consequently, reputation plays a much bigger factor in commercial success. However, the rise of TikTok, Vine, and other social media applications has allowed people to circulate their creations more easily, and to rise from obscurity.

Social media culture has arguably reduced the overall attention span in our society. Due to the reduced attention span, underground artists post short snippets of songs on social media apps to gain more mainstream attention. Snippets on promotional pages such as Worldstarhiphop, GRM Daily, Mixtape Madness and other regional platforms are centralized resources that post artists' work on their social media channels for promotional pay to wide audiences. Additionally, through our personal experiences we have noticed that catchier songs seem to have a greater likelihood to capture the attention of a younger demographic in short social media snippets on apps such as Vine, Snapchat, and TikTok. This effect of duration becomes noticeable in **Figure 13**, where the duration of a song becomes more important in deciding whether or not a song is successful from the 2000s to the 2010s. Many artists even attach dance choreography and dance challenges which help grow exposure and create culture around songs that can define eras. Soulja Boy's "Crank Dat", Drake's "Tootsie Slide", and 2 Milly's "Milly Rock" are all examples of dance crazes that did well in the music charts, defined childhoods, and evoked emotions that will be remembered by their listeners. One can define a "catchy" song as one that includes high danceability, and from our data we noticed that in fact, danceability rose in importance throughout the years. Danceability was the 8th most important feature/determinant in the success of a song in 2000, while in 2010 it rose to the 5th most important feature shown in **Figure 13**.

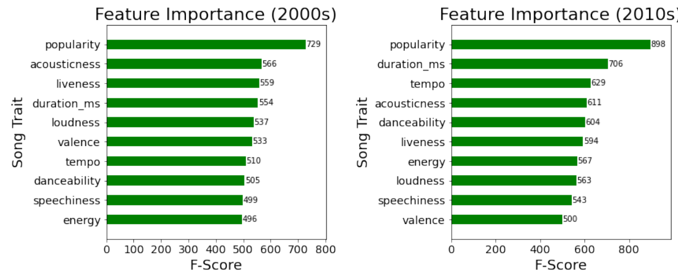

Figure 13: *Feature Importance Across Decades:* These graphs exhibit how dynamic music trends are and how features that were important in determining whether a song was successful or not change over decades. All features minus popularity were shuffled around from the 2000s to the 2010s.

Social factors relating to the reputation of an artist have been proven to have a direct impact on the "success" of an artist's music. Unfortunately, death has been a prominent part of the music industry in 2020 as well as the rest of the world. Musicians, especially those who were killed and died in unnatural ways, have achieved mainstream success and receive more attention after death. In 2020, the success of posthumous projects from famous artists such as Pop Smoke and Juice WRLD have shown that death is the biggest form of marketing on the earth. The following graph from *The Economist* (**Figure 14**) compares the commercial success of artist's ante mortem albums to their posthumous projects. The Album-Equivalent Units (AEU) are noticeably higher for the posthumous albums of all these artists. Juice WRLD and Pop Smoke's posthumous albums charted seven and four times more AEUs, respectively, than the average releases when they were alive. We can conclude that the complex emotions that death revokes leads many listeners to listen to artists for the first time and to even becoming avid fans.

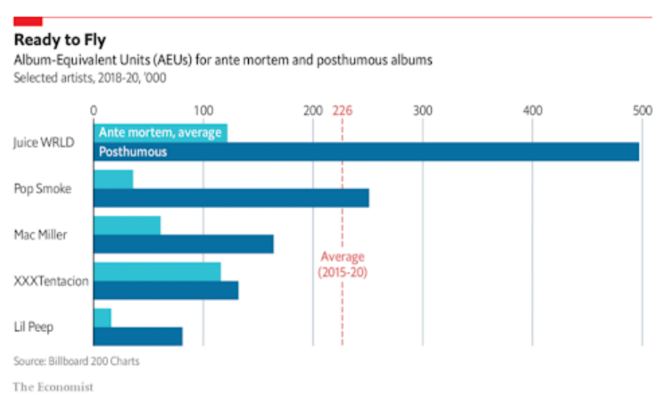

Figure 14: *Death Sells:* This figure taken from an article in *The Economist* [8] shows how the album sales of hip-hop artist deaths have been affected by their untimely deaths.

5.2 Constraints

The process of collecting, analyzing and modeling our data was very insightful, and we learned more about specific constraints due to the structure of the project. These constraints made us realize some complexities within music that affected our results. One constraint that appeared in our project was how song covers could skew our data and results. Covers can be defined as songs where the different artists sing or rap the same lyrics of another artist's song and production. Thus, according to the model, one would expect all covers of successful songs to chart well because they share all the same features, but obviously most songs covers are less successful due to the lesser reputation of the artists. Lastly, our data analysis and model implementation on this project was limited to < 4 months. Unfortunately, it's not a longer 6 month or 2 year long initiative where we have time to further learn more about the nuances that lead music to be successful and try new methods to develop a better model.

5.3 Weapon of Math Destruction?

When developing a machine learning model, ethics will enter the conversation if the model will be used in society. If this is the case, a predictive model runs the risk of causing harm to society, or influencing the prediction it is predicting on, causing a self-fulfilling prophecy. This situation is discussed thoroughly in Dr.Cathy O'Neil's book, Weapons of Math Destruction, which is also the term coined for the situation.

The predictive model of this project has the capacity to become a weapon of math destruction if used improperly. Imagine if we were able to develop a model that could predict the success of any song with 100% accuracy using the features discussed earlier. Not only would we be able to influence the music industry and reap the rewards of that, but wouldn't we also have the capability of ruining the integrity and creativity of the art? Music is usually created from the expression of emotion due to one's life journey and lessons. Making music a quantifiable measurement based on its components seems to separate the art from its origins. On the other hand, one can argue that this accurate model would also be able to promote and further assist aspiring artists in order to monetize their talent and gain a leg up in life.

6 Conclusion

Overall, our model performed very well given the nature of the classification problem at hand. We were able to achieve an accuracy of 88.8% and a recall of 53.3%. Our model was able to correctly classify a song as unsuccessful 95% of the time, while also being able to correctly classify a song a successsful 53% of the time, which are better odds than a fair coin toss. While this model alone currently may not be able to earn a record label millions of dollars in revenue or propel an up-and-coming artist to next year's Grammy's, this model can be used to supplement the opinion of someone with a good ear for music such as a label's recruiter. This, along with improvements in accuracy and recall through the addition of more complex features would allow this model to be valuable to someone in the music industry.

One of the biggest takeaways and affirmations from this project is the concept that music is innately a communal and a societal art that unites people based on the emotions that it evokes. This emotion can be a range of reactions originating from the sound itself to even the reputation of the artist. Thus this makes it difficult to quantify the correlation between quantified measurements of musical attributes to the "success" of a song, when this success is also a "ranking" based on subjective ears. Music at the end of the day is a societal art and communal bond. It's cultural impact can not simply be quantified as music evokes emotions, creates memories, and has always been a unifying aspect of humanity.

7 References

1. Spotify for Developers Web API. *Spotify*, developer.spotify.com/documentation/web-api/.

2. DeFoe, Daniel. "Data on Songs from Billboard 1999-2019." *Kaggle*, 3 Mar. 2020, www.kaggle.com/danield2255/data-on-songs-from-billboard-19992019.

3. Ay, Yamaç Eren. "Spotify Dataset 1921-2020, 160k+ Tracks." *Kaggle*, 25 Nov. 2020, www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks.

4. Pennington, Jeffrey. *GloVe: Global Vectors for Word Representation,* nlp.stanford.edu/projects/glove/.

5. Trust, Gary. "The Longest-Leading Billboard Hot 100 No. 1s." *Billboard*, 2 Apr. 2019, web.archive.org/web/20180405073449/www.billboard.com/articles/columns/chart-beat/6077132/longest-leading-hot-100-no-1s.

6. Tianqi Chen and Carlos Guestrin. "XGBoost" *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016,* doi:10.1145/2939672.2939785.

7. "Chapter 14: Support Vector Machines." *Hands-on Machine Learning with R*, by Bradley C. Boehmke and Brandon Greenwell, CRC Press, 2020.

8. "Posthumous Albums by Young Rappers Are Topping the Charts." *The Economist*, The Economist Newspaper, 23 July 2020, www.economist.com/graphic-detail/2020/07/23/posthumous-albums-by-young-rappers-are-topping-the-charts