INTRODUCTION

In the music world, it is a pretty tough game for an artist to get discovered given the low barrier to entry. According to the Music Times (2014), there is a discouraging percentage of artist being commercially successful, with a classification break-down of “0.2 percent of artists as *mega-stars*, 0.9 percent as *mainstream*, 1.4 percent as *mid-sized*, 6.8 percent as *developing,* and the whopping 90.7 percent as wholly *undiscovered*”. Behind the scenes, it is always the case where musicians, music company executives, or even friends and family members of the artist are debating whether a song could be a hit or not. What lies in this spectrum is essentially the prediction accuracy of a song’s commercial success.

Long had it been a tradition that the music industry being frenzied by Artists and Repertoire (A&R) guys that are constantly seeking for talented musicians and songwriters that they believe could have the potential to create the next hit. More often than not, what they are doing poses a huge risk to the music companies they are tied to. Once they found an artist that they believe could rise to stardom, as per the IFPI (Weibe 2018), the music company will proceed by allocating investment dollars into the artist, usually in the range of $500,000 to $2,000,000 for a newly signed artist. This is an extremely huge risk for any company to pay for an act that is not proven and, in general, the chances of turning into fruition are slim. That said, getting a song to be a top hit is very important these days as it means more revenue for the artist and the music management company. However, the main problem with A&R’s prediction in a song’s success is somehow rooted in personal character and traits, where biases, emotions, and industry connections usually determines if an artist gets signed or not.

The problem I am considering is to reduce the risks and cost associated with A&R and the music company while increasing the chances of a song appearing in the Billboard Hot 100 Chart based on song and artist characteristics. In the music industry, the Billboard Hot 100 is usually the benchmark in determining a song’s commercial success, so I will be using the Billboard Hot 100 as a benchmark for this project. According to Molanphy (2013), in order for a song to achieve the Billboard Hot 100, the song “must be placed well in all three categories: sales, airplay and streaming”. Therefore, I am interested in developing a model to help artist and music companies see if a song can potentially achieve the Billboard’s Hot 100 list or not.

RELATED WORK

There were a couple of groups that have attempted this problem before. The first one is by *Mohamed Nasreldin, Stephen Ma, Eric Dailey, Phuc Dang,* who managed to have an

accuracy rate of 68% with their highest performing model, which is the XGBoost (Nasreldin, 2018). They also ran different models such as random forest, KNN, decision trees, logistic regression, and support vector machines to compare results. In addition, another group of scientist at the University of California, Irvine, who did a similar study, claimed to have a much higher prediction accuracy of 86% (Bill Murphy, 2018). The scope, however, is to look at songs that were placed in the Top 100 Singles Chart in the United Kingdom for at least a week. Their study was conducted with roughly 500,000 pop songs ranging from the year 1985 to 2015.

DATA DESCRIPTION

My data for this problem is from the MillionSongDataset, which was the "largest music activity dataset made available to researchers" at that time (Brian, 2011). Out of curiosity about how and why did the owners created such a dataset, I decided to do a thorough research on their website at millionsongdataset.com (Million Song Dataset). I found out that the owners of the dataset used The Echo Nest API and some information from a local copy of the *musicbrainz* server to create the dataset. Also, I get to know that their purpose of putting together this dataset is not only "to encourage research on algorithms that scale to commercial sizes and to provide a reference dataset for evaluating research, but also as a means of a shortcut alternative to creating a large dataset with The Echo Nest's API and to help new researchers get started in the music information retrieval (MIR) field." (Bertin-Mahieux at.el, 2011).

However, the dataset I am using is a smaller scale of the original dataset. The samples have been trimmed significantly, from a roughly 280GB 1 million samples and 55 variables dataset to a tiny fraction consisting of around 10,000 samples and 23 variables. I decided to use this dataset as it contains some of the most important musical factors as determined by the US National Science Foundation's GOALI - Grant Opportunities for Academic Liaison with Industry program (Bertin-Mahieux at.el, 2011). Variables that I deemed important to this specific project are variables such as artist hotness, artist familiarity, and some other artist and song characteristics.

The dataset has 23 variables in total. I started my preprocessing steps with removing some of the variables that I deemed useless for this project, such as 'artist_id' and 'title'. I then proceeded with some initial data visualization to spot missing values as shown in (*Appendix A*), which I ended up removing four variables that have missing values - 'song_hotness', 'artist_longutude', 'artist_location', and 'artist_latitude'. After cleaning the dataset, I left with 15 variables that were used in the model. As for the 'artist_familiarity' variable, I decided to just

impute the four missing values instead of removing the entire column because it is just four out of 10,000 observations. I then split my data into train and test dataset to a ratio of 80% to 20% respectively. Also converted the response variable 'BillboardHot' into a factor. The pre-processing steps taken are shown in the process chart below.

METHODS

In this project, the method I used was the XGBoost. In similar projects attempted by the other two groups as mentioned above, a group found the best results came from XGBoost as well. Therefore, I am pretty optimistic that XGBoost is the right method to use here. With XGBoost, I was able to tune certain areas specifically to my likings. For example, I tuned the ETA at 5 different levels (*see discussion*), and I was able to find the one that returned the best outcome. I also did tuning for the rest of the XGBoost spectrum, finding the best overall result in the process.

For the strength of this model, I realized that XGBoost is fairly easy to conduct, despite the long running time on certain chunks of code. Its flexibility in tuning is also something that I cherish as it improved the model's meaningfulness to the project even further as compared to without tuning. Also, this model will not expire or run out-of-date, as new methods and applications come along it will only improve this model. Therefore, the core concepts can be developed once, and use for a long period of time with some updates here and there. Further, its suitability to the problem lies in the case where it takes no biases unlike human beings. The model predicts based on matching components within its model. I think this is by far one of the most important strength of the model. That said, it could lower various risk factors for music companies and provide a much more secure prediction as opposed to a human predictor.

On the flip side, the biggest weakness of the model is that it is not practical to certain genres. Take the Rap genre for example, which is usually measured based on its lyrics and beats instead of the other factors that constitutes Pop music. A group of scientist from University of California, Berkeley (Abraham, Tony, et al.) studied specifically on Rap music and what it takes for them to get listed on the BillboardHot 100 Chart. They managed to garner an accuracy rate of over 70% with their model. While my model focuses more on the Pop genre, if those

guys at UC use their model to predict Pop music, I believe they will get a different outcome. Likewise, the same applies to my model when predicting Rap music based on Pop components. Therefore, I would say that my model is not suitable and not a good predictor for all genres of music, and for this reason, it is part of the model's limitation.

RESULT & DISCUSSION

From my very findings in the beginning before running the XGBoost, it looks like artist_familiarity (*Appendix B)* and artist_hotness (*Appendix C*) are likely candidates that will propel a song towards the appearance in the BillboardHot 100 chart. That was before running the XGBoost model. However, as I proceeded with the XGBoost, I started with training the dataset and what I found was that when the AUC increases, the error rate tend to decrease as shown in (*Appendix D).* The result suggest that when the AUC is 0.66 the error rate is at 0.11. But as the AUC is increases to 0.99, the error rate drops significantly to 0.03.

To better improve the model, I knew I needed to do some tuning. So I started my tuning inside of the XGBoost model and continued with the max depth and min child weight. The best result came in when Max Depth is at 5 and Min Child Weight is at 10. After that, I did a gamma tuning and added more rounds just so that it brings more value to the method. I increased the number of rounds from 100 to 1000 to see how it perform. The result suggest I use 62 trees. I then proceeded with tuning the subsample and colsample_by_tree. It returns the best results when subsample is at 0.9 and colsample_by_tree is at 0.8, though it does looks like it does significantly better with more samples. It may mean that there are a bunch of variables that do not provide high magnitude of predictive power so those trees that were built with them are bad predictors. In addition, I tuned the ETA on 5 different levels, which are 0.3, 0.1, 0.05, 0.001, and 0.005. I then plot them as shown in (*Appendix E)* to see how they look. The best one appears to be when the ETA is at 0.1.

From there, I implemented all the tunings done into the final model. I first did it with a usual cut-off rate at 0.5, and that yielded me a balanced accuracy rate of 0.51346. I was thinking to increase the accuracy so I decided to lower the cut-off rate from 0.5 to 0.1. Thankfully, the balanced accuracy has increased from about 51% to 55%. To find out if there will be any imbalanced data, I ran another round of code and that actually returned a slightly lower balanced accuracy rate of 53.7%, but that's ok, I will take that.

Hopping on from there, I ran a variable importance with XGBoost. The results can be seen in (*Appendix F)* For some reason, it aligned with my initial exploration that 'artist_hotness' would be the most important variable. 'artist_familiartity' however, came in at fourth, though not

far off as the top four variables importance are separated by just less than 0.1 when taking margins ratio into consideration and most likely fall somewhere between the range of 0.13-0.18. To see if these are really the most important variables, I decided to try out a SHAP variable importance. The results are shown in (*Appendix G*). To my surprise, 'duration' ended up being the most important variable followed by 'start_of_fade_out', while 'artist_hotness' came in at third. Also, something I find interesting is that 'duration' is leading by a huge margin ratio to the second most important variable.

To further enhance the SHAP value and its effect on the model output, I plotted a SHAP feature value (*Appendix H*). So it looks like 'duration' is the one that leads the pack with the highest SHAP value. However, the feature value of 'artist_hotness' stretches out the highest in terms of its feature value. Also, I would ignore 'year' here because every year songs are being created and I don't think this would be the appropriate way to measure the 'year' variable as it provides no sensible meaning in this context.

CONCLUSIONS & FUTURE WORK

In conclusion, the final result provided a balanced accuracy rate of 53.7%. Overall, the results have shown that song and artist characteristics does play a role. To apply into the real world, this model provided the insights that can be implemented by artists and music companies to increase their chances of achieving the Billboard Hot 100 status. It will also help lower the risks and money factors for music companies. However, something to note is that at the end of the day, there are only 100 slots available and with the growing number of artist and singles that's being released every year, the competition is only getting tougher as the years passed.

Given more time and resources, I would dig deeper and include a larger number of variables such as artist gender, age, years of music training experience, genre, song complexity, budget, label and publishing deal status to have better supporting data variables behind the overall model results. In addition, I will look into implementing other machine learning methods such as random forest, logistic regression, and decision trees to get a better accuracy rate since my XGBoost model accuracy rate is being a little subpar at just 53.7%. Lastly, I will use a more recent and up-to-date dataset to reflect industry compatibility.

BIBLIOGRAPHY

Abraham, Tony, et al. "A Data Science Exploration of Rap Lyrics and What It Takes to Make It onto the Billboard Charts." *R.A.P.: Rap Analysis Project*, UCBerkeley.edu, people.ischool.berkeley.edu/~nikhitakoul/capstone/index.html. Accessed 20 Nov. 2020.

Bertin-Mahieux, Thierry, et al. *THE MILLION SONG DATASET*. 2011.

Bill-Murphy, Jr. "Scientists Studied 504,810 Songs. Now They Can Predict a Hit With 86 Percent Accuracy." *Inc.Com*, 17 May 2018, www.inc.com/bill-murphy-jr/scientists-studied-504810-songs-now-they-can-predict-a-hit-maybe-before-its-even-recorded.html. Accessed 29 Oct. 2020.

Brian. "Taste Profiles Get Added to the Million Song Dataset." *The Echo Nest Blog*, The Echo, 27 Oct. 2011, blog.echonest.com/post/11992136676/taste-profiles-get-added-to-the-million-song. Accessed 29 Oct. 2020.

"Million Song Dataset." *Millionsongdataset.com*, millionsongdataset.com/. Accessed 29 Oct. 2020.

Molanphy, Christ. "How The Hot 100 Became America's Hit Barometer." *Npr.Org*, 1 Aug. 2013, www.npr.org/sections/therecord/2013/08/16/207879695/how-the-hot-100-became-americas-hit-barometer. Accessed 29 Oct. 2020.

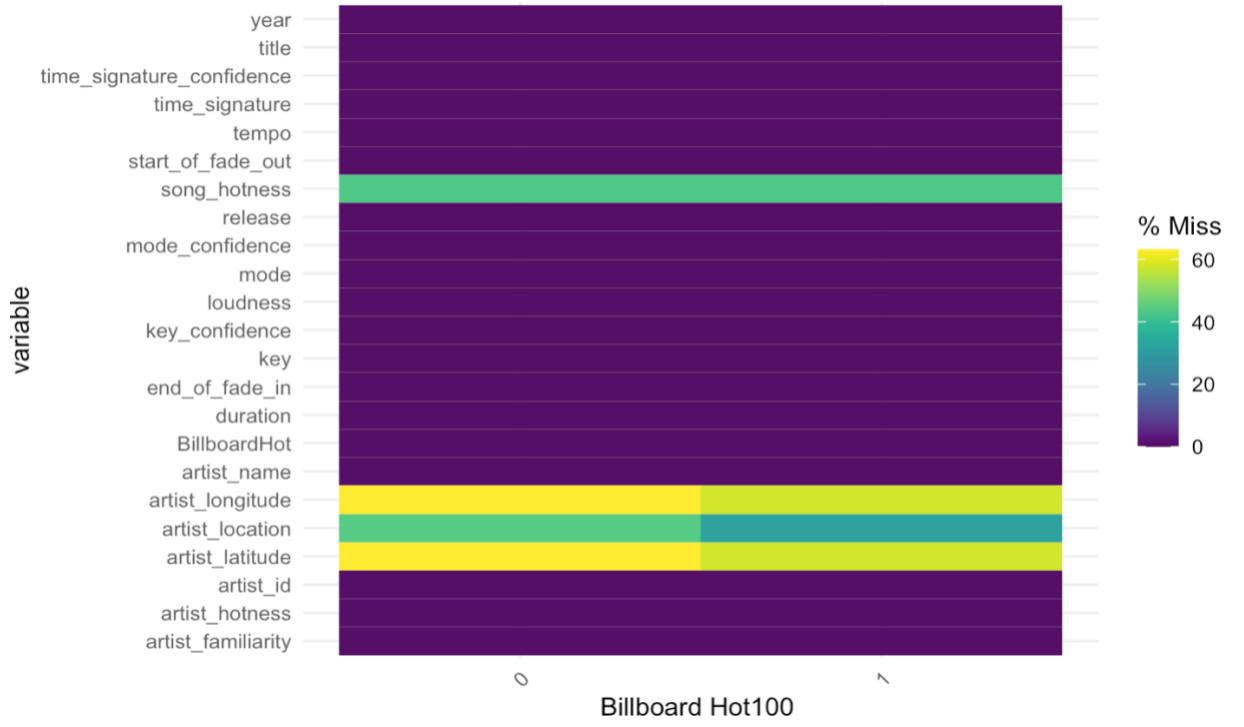
Music Times and Menyes, Carolyn. "If You're a Musician, Chances Are You're Totally Undiscovered, Says a New Study." *Music Times*, 21 Jan. 2014, www.musictimes.com/articles/3563/20140121/youre-musician-chances-totally-undiscovered-new-study.htm. Accessed 29 Oct. 2020.

Nasreldin, Mohamed. "Song Popularity Predictor." *Medium*, Towards Data Science, 5 May 2018, towardsdatascience.com/song-popularity-predictor-1ef69735e380. Accessed 29 Oct. 2020.

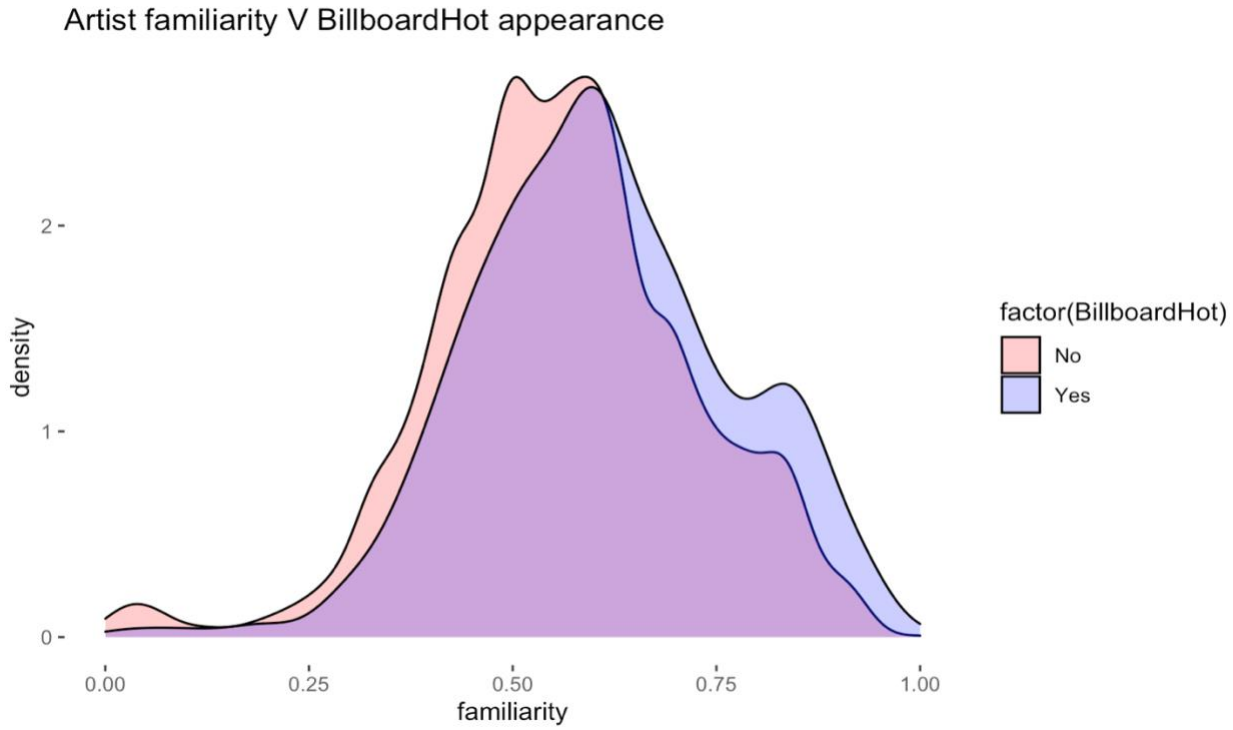
Wiebe, David Andrew. "How Much Advance Do Record Labels Give, And How Much Should You Try And Get?" *Music Industry How To*, 17 Jan. 2018, www.musicindustryhowto.com/how-much-advance-do-record-labels-give-and-how-much-should-you-try-and-get/. Accessed 20 Nov. 2020.

APPENDICES

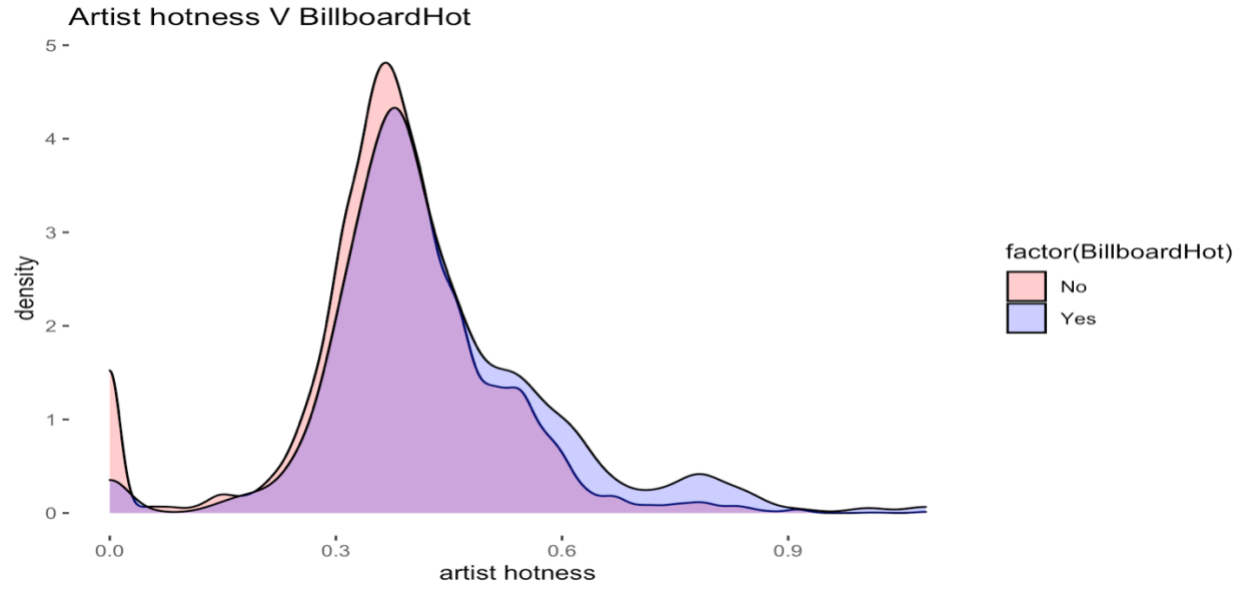
APPENDIX A – Graphing Missing Values



APPENDIX B – Exploratory plot



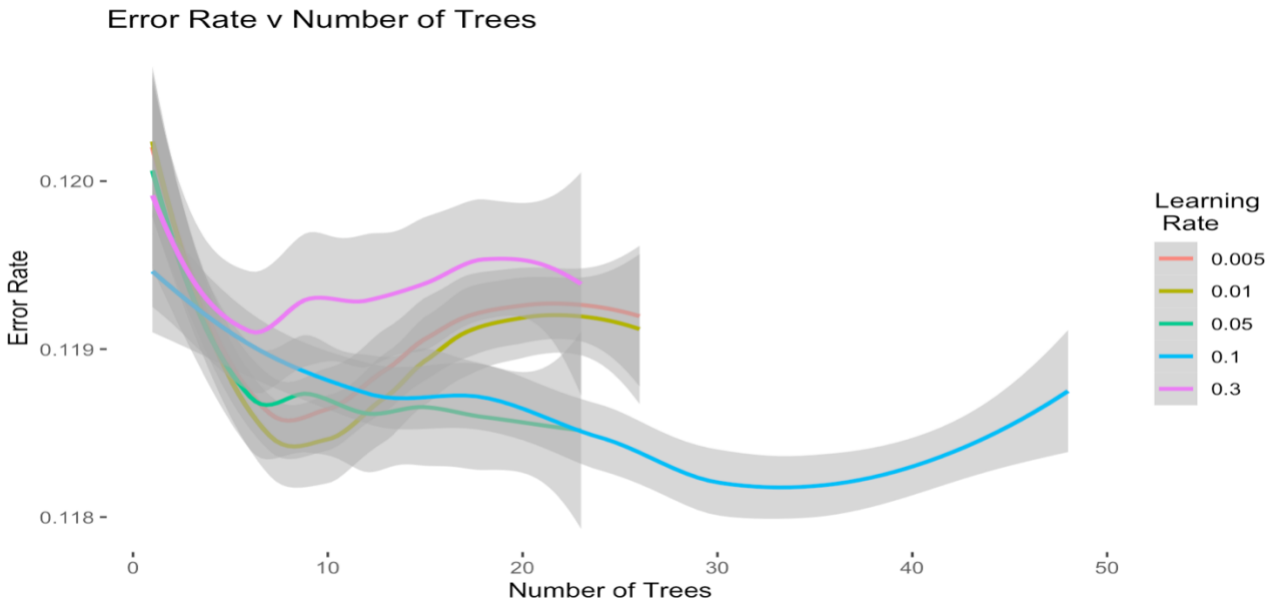
APPENDIX C – Exploratory plot



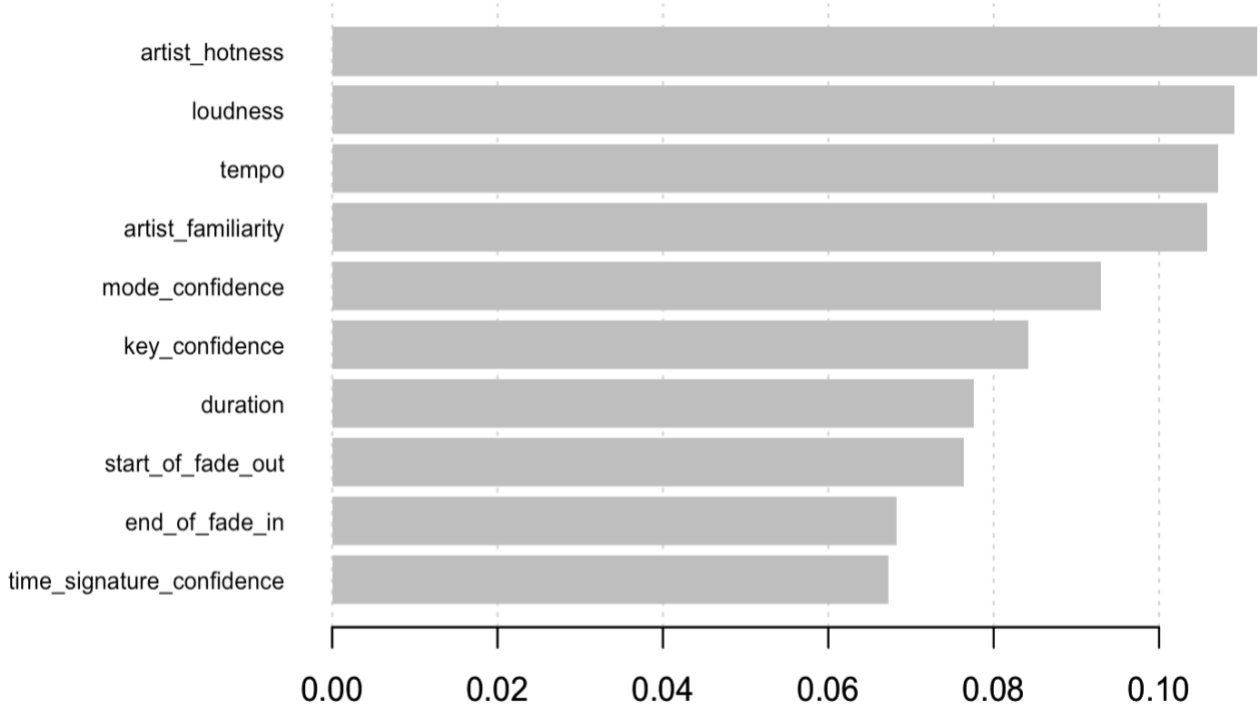
APPENDIX D – First output with XGBoost

```
[1]     train-auc:0.661142      train-error:0.111986
[21]    train-auc:0.900567      train-error:0.104987
[41]    train-auc:0.957830      train-error:0.092238
[61]    train-auc:0.985563      train-error:0.071491
[81]    train-auc:0.995863      train-error:0.051244
[100]   train-auc:0.999135      train-error:0.033996
```

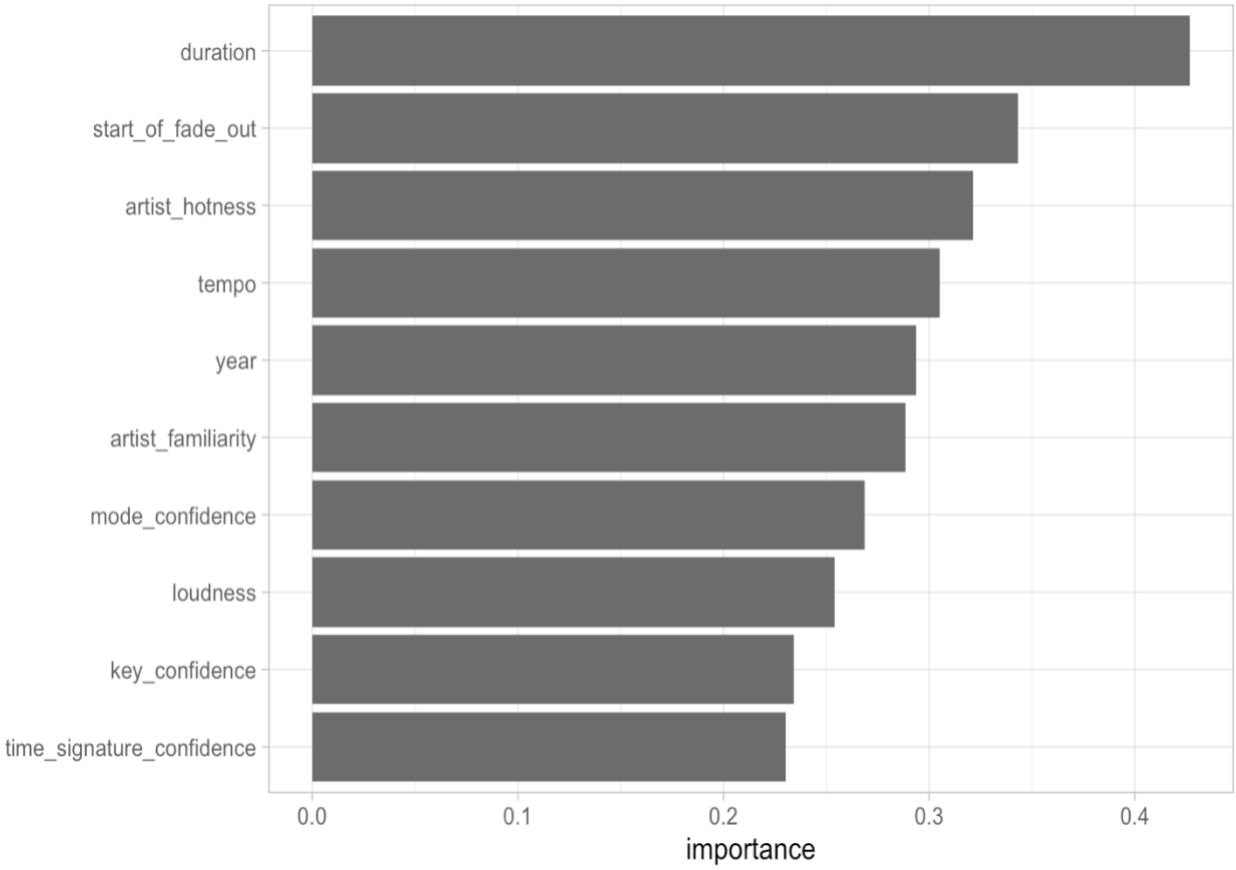
APPENDIX E – ETA Tuning



APPENDIX F – XGBoost Variable Importance



APPENDIX G – SHAP Variable Importance



APPENDIX H – SHAP Feature Value