Predicting the Popularity of Songs

Michelle Dang, Lisa Wang, Victoria Yuan
{mbdang, x766wang, csyuan}@uwaterloo.ca
University of Waterloo
Waterloo, ON, Canada

Introduction

This study tackles the problem of determining how commercially successful a given song will be in the music industry purely using attributes of the music content itself. A song will be classified as a popular hit if it has appeared on the Billboard Hot 100 list for multiple weeks. In the music industry, there are usually countless factors, including external ones such as artist reputation, economy, social influences, etc., that contribute to the popularity of a song but this study will focus exclusively on audio features of hit tracks. This paper aims to highlight some of the main reasons why these audio features are appealing and showcases some frequently seen patterns in the data. Due to the fact that commercial success is something that many artists, record labels, and producing companies are trying to achieve, this study could potentially pique the interest of those who are involved in the industry. The results found in this paper could potentially also be used to predict how popular certain songs will become in the future.

Contributions

The popularity of a song may be affected by many external factors such as artist popularity or advertisement spending as well as internal factors such as the content of the song itself. An accurate hit song prediction algorithm could be useful for industry professionals to predict the popularity of songs before their release or reveal ways of modifying a song to optimize its chance of becoming a hit. We aimed to solve the problem of how to classify a hit song solely using the content of the song through low-level and high-level acoustic properties.

We used the Billboard Hot 100 chart to determine if a song is considered a hit, as well as the Million Songs Dataset to construct our training and testing set, along with each data point's acoustic properties. Each property is associated with a numerical value. Out of those provided by the dataset, we considered tempo, loudness, duration, key, and the time signature. With the training set, we implemented a decision tree using the ID3 algorithm and iterated on the solution by fine-tuning the split points of the acoustic properties. Due to the many internal and external factors that affect the popularity of a song, we had anticipated that the results would not be completely accurate. In fact, our highest accuracy was found with a decision tree of maximum depth 8, which produced an accuracy of 67.50% on our test set. Using this decision tree, we found that hit songs followed general song trends that hold true to this day. For example slower, quieter songs and louder, faster songs were generally popular. However, caution is advised on these over-generalizations as it is certain that external factors play a sizable role in determining a song's popularity.

Related Work

Previous work on the topic of predicting song popularity has examined both internal factors such as song content as well as external factors like artist popularity on social networks and social context of songs. In the study "What makes a music track popular in online social networks?" [2], all of these factors were studied using songs extracted from popular music social media site Last.fm. While low level features in song content such as tempo and melody accounted for 70% of the algorithm's accuracy, artist reputation and social context were found to be much less important (Ren, Shen, and Kauffman 2016). In addition, a comparison of several algorithms was done and Bagging was found to be the most successful one when performed on the song content factor.

Instead of measuring popularity within social networks, Yang et al. [3] defined song popularity as a score from the product of its play count and number of listeners (both in log scale). They used CNN models to generate high-level features of songs and trained both shallow and deep neural network models to predict these scores. Their study concluded that DNN models were more accurate than shallow for predicting the popularity of Western songs, however they only examined songs from within one year to avoid changes in public music taste (Yang et al. 2017).

In contrast, the methodology taken in paper "Hit Song Science Once Again a Science?" [1] was much simpler and did not involve high-level feature generation. Again, low level features such as tempo, time signature, song duration, and loudness were analyzed using a perceptron algorithm (Ni et al. 2011). Meanwhile, in "Hit song prediction: Leveraging low- and high-level audio features", a paper by Zangerle et al [4], they had used both low- and high-level audio features with a DNN model to predict the peak rat-

ing of a song. External factors were not included and focus was given on jointly exploiting low- and high-level audio features, being the first research conducted to examine both types of features concurrently. They also factored in mood and vocals as they are crucial aspects of determining whether songs are disliked. They found that the combination of the two features allowed for improvement of prediction performance and by incorporating the release year, as a high-level feature, reflected musical fashion and trends (Zangerle et al. 2019).

When taking in the findings of previous work, they had all approached the problem in various ways. Some of them examined both internal and external factors while some examined both, and they consistently used different methodologies as well. However, a commonality was that the majority included low-level features as their input features. This paper will do the same, not only because they are the most notable attribute in identifying a hit song, but they are also easily accessible through the open source Million Songs Dataset.

Finally, as observed in several of the related works, trends change greatly over time in the music industry, resulting in very different characteristics in hit songs. In fact, Zangerle et al saw that audio features previously relevant for success changed at a rapid rate, indicating that musical fashion is short-lived. As an attempt to analyze a window of time in order to accurately reflect these evanescent trends, the data used in this study will only contain songs from years 2000 to 2010.

Methodology

Our problem will be solved using decision trees generated by the ID3 algorithm and we will be training and testing using data from the Million Songs Dataset and the Billboard Hot 100 chart. As discussed previously, all the data will be taken from the years 2000-2010 as an attempt to confine the musical trends that are studied. The evaluation of our algorithm will measure the percentage of correctly classified songs from our test set.

Evaluation Method

The Millions Songs Dataset (MSD) will be used in this project, which is a dataset containing feature analysis and metadata of one million songs, with original data contributed by The Echo Nest. This dataset was chosen not only as it is a freely-available dataset, but it already contains desirable features for evaluating the popularity of a song in this project. In addition, it has been widely used in previous AI research projects involving music. The fields for each song includes basic metadata such as artist, title, year of release, artist location, etc. and audio-focused features like key, tempo, time signature, as well as audio analysis arrays. Since the full dataset is over 300GB, a summary HDF5 file of the whole dataset which has size 300MB, will be queried from instead. The summary HDF5 file contains all the same metadata but no arrays containing information such as the audio analysis, similar artists, and tags. This dataset only contains songs up to 2011, which is why song selection will be limited to those released between 2000-2010. Within this subset of songs from MSD, the selection will be refined to 1000 data points, with an even split between "hit" and "non-hit" songs. "Hit" songs will be classified as songs that appeared on the Billboard Hot 100 chart over a threshold of a certain number of weeks. This quantity of weeks will be experimented with during our evaluation to see whether or not a certain threshold will yield higher accuracy results and reduce the possibility of false positives. This will make the distinction clearer between a very popular song that has appeared for many weeks versus a song that might have barely made the chart for one week by chance.

To retrieve the songs that have been on Billboard Hot 100, a publicly available Python API[1] will be used to get all charted songs between 2000-2010. Furthermore, the week attribute of the chart entry will be used to determine if a song is a "hit" as described above. Half of the training set will be selected among the intersection of the results from Billboard Hot 100 and the MSD and the other half among the subset of MSD that are not considered "hits." For the test set, 400 data points distinct from the training set will be chosen from the remaining subset of the MSD. It was found that out of the one million songs, 308 515 songs were released and 3992 distinct songs had charted on Billboard Hot 100 between our specified time period of 2000-2010. Since randomly selecting data points for the test set would likely yield a majority, if not all, of the songs to be non-hits, the decision was made for the test set to be composed of 200 hits and 200 non-hits, distinct from the training set.

The algorithm will then be evaluated based on the percentage of songs predicted correctly. With respect to the timeline, one week will be given to pre-process the data and two weeks to implement the algorithm. The remaining week will be used for adjustments and evaluations, with more detail in the following section.

Algorithms

The classification algorithm that was chosen was a decision tree constructed using the ID3 algorithm, due to both ease of human interpretation and algorithm simplicity. In addition, the data that was used in this study are exclusively tabular real-valued numbers, which could potentially work better with a decision tree rather than a neural network. By using a decision tree, there is more flexibility involved since the analyzed features, as well as their split points, could be easily adjusted to prevent overfitting. The adjusted decision tree is also simple to re-generate using the program that will be built.

In the Algorithms portion of the study, there will be 3 major tasks involved. First, given the training set, optimal split points will have to be calculated by comparing the level of information gain for possible split points. Since the ID3 algorithm will be followed closely, to determine if a point is a possible split point, all points where the feature value changes from X to Y will be considered. If in this set there exists two points, one where the feature takes value X and the other value Y, where the label values are different, then

[1] billboard.py of https://github.com/guoguo12/billboard-charts

this point will be kept for consideration. Finally, the expected information gain from all potential points is calculated by splitting the examples into 2 subsets and the one that yields the maximum information gain is picked. Our tree will be a binary decision tree as for every split point for each feature, we will split the examples into 2 subsets.

The second major task is to determine an optimal ordering of the features, which is calculated by again maximizing the amount of information gain. This will ensure that the generated decision tree has the least number of levels and nodes, which will make the classification process cleaner and easier for the algorithm. To do so, the uncertainty of before and after testing the feature is calculated to see how much information can be gained. The uncertainty is calculated by analyzing the distribution of the results through the entropy calculations. At every node of the tree, the feature that results in the highest information gain is picked, even if it has been picked at a previous node. This encourages the tree to make the decisions that make the greatest difference in the classification as quickly as possible.

The last task will be to construct the actual decision tree using the information that was determined from the previous two tasks. There are 3 base cases for node construction: the case when all the remaining examples are in the same class, when there are no features left, and when there are no examples left. When all the remaining examples are in the same class, a node with the unanimous classification is created. When there are no features left, indicating that the data is noisy, we create a node with the majority classification. Finally when there are no examples left, then there are no previous examples in the training set with this set of features, so a node is created with the majority classification at the parent node. Otherwise, we recursively build the rest of the tree by choosing the optimal feature and its optimal split point. At the end of this algorithm, a full tree will be built which is able to classify the training data perfectly, but may overfit on the test data.

Pruning may have to be performed in order to fix the problem of overfitting. We will prune according to 2 different criteria: maximum depth and minimum information gain threshold. The optimal pruning values for each of the criterion will be determined by comparing the test accuracies. In addition, more complicated pruning strategies will also be investigated by combining different combinations of both maximum depth and minimum information gain threshold to see if these trees are able to perform better. Several different test datasets will also be generated to compare average test accuracies of each pruning strategy across different sets in order to obtain the optimal decision tree.

Each member of the team will work a different portion of the implementation for an estimation of 10 hours. One team member will mine and format all relevant data needed and construct both the training and test sets. One team member will construct the full tree with the training set using the algorithm methods stated above. One team member will implement different pruning strategies to maximize the prediction accuracy on the validation set. If time permits, all members will be responsible for generating additional test sets so that we can run more data through our trees and get a more holistic view of which tree performs the best.

Results

The full decision tree generated by our algorithm had a prediction accuracy of 100.00% on the training set and 57.25% on the validation set. It has a depth of 29 and overfits on the training set which leads it to perform poorly on the validation set. In order to fix the problem of overfitting, we then investigated the performance of trees that underwent a variety of pruning strategies.

The preliminary results of our implementation show the prediction accuracy of a decision tree that has been pruned to a maximum depth as well as pre-pruned and post-pruned to a minimum information gain threshold and some combinations of the three pruning options.

After a full decision tree was generated, maximum depth pruning was implemented for depths of 5-14 (Figure 1). We omitted depths of less than 5 and greater than 14 because they did not result in better results or were equivalent to the full tree. The results of pruning by maximum depth showed that the best results on the validation set were at depth 8, where the prediction accuracy was 67.50%.

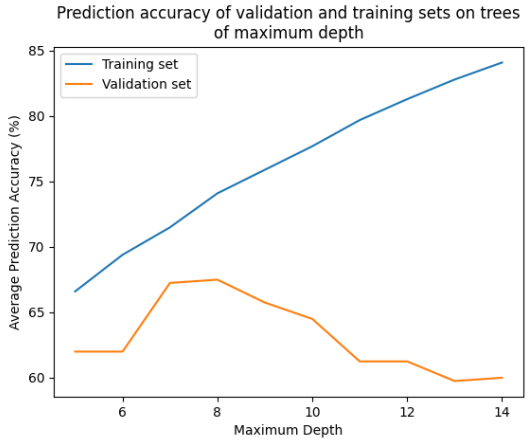

Figure 1: Prediction Accuracy of Validation and Training Sets of Trees of Maximum Depth

Using the same full decision tree, we also implemented post-pruning to a minimum information gain threshold (Figure 2). The range of thresholds tested was between 0-1.5 in 0.1 increments. The results of pruning by these thresholds showed that the best results on the validation set was at 1.1, with a prediction accuracy of 59.00%.

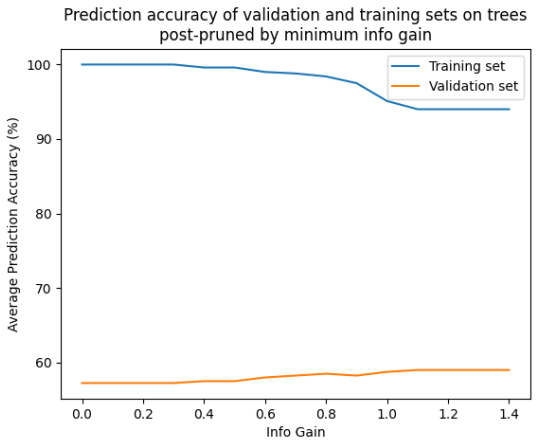

Figure 2: Prediction Accuracy of Validation and Training Sets on Trees Post-pruned by Minimum Info Gain

We also implemented pre-pruning to a minimum information gain threshold and generated the tree using thresholds in the range of 0-0.15 in 0.01 increments (Figure 3). We found that the threshold with the highest prediction accuracy on the validation set was the threshold of 0.02 at 61.75%.

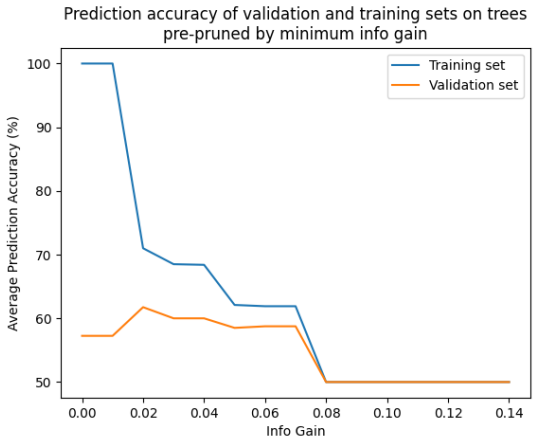

Figure 3: Prediction Accuracy of Validation and Training Sets on Trees Pre-pruned by Minimum Info Gain

Finally we implemented a combination of maximum depth pruning and post-pruning by a minimum info gain and found that the tree at maximum depth 8 and no minimum threshold performed the best. Similar results were found for the combination of maximum depth pruning and pre-pruning by a minimum info gain. We have concluded that the combination pruning did not increase the validation accuracy of the decision tree.

Our results recorded the trends from changing the minimum info gain and maximum tree depth in comparison to our full decision tree with 57.25% prediction accuracy. As we increased the minimum info gain threshold while post-pruning, the prediction accuracy increased slightly but plateaued at 59.00% accuracy. The trend as we increased the minimum info gain threshold while pre-pruning was a slight increase in prediction accuracy for values 0.02-0.07 but a sharp decrease for threshold values greater than 0.07. Lastly, the trend observed as we increased the maximum tree depth from 6-14 was a sharp increase in the prediction accuracy on the validation set from depths 6-8 and a steady decrease for depths greater than 8.

From the results above, we found that the decision tree which performed the best was the tree with a maximum depth of 8 at a validation set accuracy of 67.50%. The decision tree with the worst performance on the validation set was the tree post-pruned by minimum info gain at 59.00% accuracy. Both experiments performed better than the full decision tree, which was expected as the full tree was overfitted to the training data.

These results are similar to what we would have expected based on our training data and the results of prior related works. In the study "What Makes a Music Track Popular in Online Social Networks?", they obtained a similar result of 70% prediction accuracy when using only the acoustic musical content of songs to classify hits. However, their dataset used songs from a smaller time range (2005-2013) and they were able to increase the prediction accuracy to 80% by considering additional factors such as social media presence and artist reputation (Ren, Shen, and Kauffman 2016). On the other hand, our results spanned a greater time range of 10 years and only considered low level acoustic properties. It is possible that the larger range of popular song trends to consider and lack of external song factors may have contributed to a slightly lower prediction accuracy of 67.50%.

Conclusion and Future Work

If the music industry was able to determine how popular a song will be prior to release, based solely on its acoustic features, it would be a huge game changer in how music is composed, advertised, and distributed. Thus, much research has been done in correlating the internal and external features of songs and how they have performed commercially. However, there are many contributing factors in a song's popularity and virality; previous studies have examined different combinations of acoustic features, promotion and advertisement, artist prominence, and so forth. In this paper, we focused on acoustic features and examined the possibility of determining a song's popularity through its low-level and high-level acoustic features, exclusively.

With respect to our project, we defined a song as popular if it appeared on the Billboard's Hot 100 chart at any time. Using the Million Songs Dataset, we created a subset of this dataset which only contained popular songs and produced a training and test dataset where half of the data points were hits, and half were not. We then constructed a binary decision tree using the ID3 algorithm, which created split points among our chosen acoustic features: song duration, key, loudness, tempo, and time signature. These features were provided in the original Million Songs Dataset, which was derived by The Echo Nest. Originally, we were planning to examine danceability and energy as well, but these two features were not analyzed by The Echo Nest. We then used the decision tree to evaluate the test set and

experimented with different pruning methods, namely combinations of pre- and post-pruning on maximum depth and minimum information gain thresholds, in order to produce an optimal tree, while avoiding overfitting. The evaluation of the tree was determined by measuring its accuracy rate of classifications compared to whether the songs were actually popular hits or not.

Ultimately, the best tree found was the decision tree pruned to have a maximum depth 8 with no additional pruning, which yielded an accuracy of 67.50%. From these results, we can safely say, at the very least, that acoustic features and song popularity are correlated, and that popular songs share similar audio features. In particular, some interesting results for songs with loudness less than -14.61 and a slower tempo, between 96.75 and 96.82, were classified as hits, whereas slightly louder songs between -14.61 and -9.89 were primarily classified as hits if they were either short (less than 128.82 seconds) or longer than approximately 300 seconds. This seems to imply that in general, quieter songs at a lower tempo or with a longer duration tended to be popular. Songs with tempos either below 125 or between 131 to 156 were also broadly classified as hits, particularly those louder than -5.91. For songs with a faster tempo than 156, the decision tree favoured songs above loudness -5.31 and less than 247 seconds. In music, the most commonly used tempo marking is allegro, which is between 120-160 BPM, which also includes the alleged "heartbeat tempo" of 120-130 BPM. Our results look to conclude that many popular songs fell in between those tempo ranges and that for very fast songs, the ones that are louder do better on charts. These two particular observations fall within reason of what is generally perceived to be popular styles of music.

Given the opportunity to further work on this project, we would like to consider collecting data ourselves rather than using the pre-existing Million Songs Dataset. A bulk of our implementation and analysis was restricted due to the inhibitions of our dataset, which while convenient, was outdated as it only included songs up to 2011. In addition, the music industry and trends change quickly, and as such, the results from this study may not be practically applicable to modern-day music and may only serve to be an indicator of musical trends throughout time. Furthermore, we are unsure of the criteria used to select songs for the dataset and in addition, results may be different if we used other music charts to classify popularity instead of the Billboard Hot 100, which is the music industry standard record chart in the United States.

We would also like to incorporate external features into our project analysis as well, with social media, virality phenomenons, industry and societal trends as examples. We believe that while musical features contribute to a song's popularity, with our results showing that hit songs generally share similar acoustic features, songs are greatly influenced by non-acoustic features as well, especially in this age of global connectivity. Eliminating these external factors from consideration would only yield a one-sided approach to examining what makes a hit song a hit.

References

Ni, Y.; Santos-Rodriguez, R.; Mcvicar, M.; and De Bie, T. 2011. Hit song science once again a science. In *4th International Workshop on Machine Learning and Music: Learning from Musical Structure, Sierra Nevada, Spain*. Citeseer.

Ren, J.; Shen, J.; and Kauffman, R. J. 2016. What makes a music track popular in online social networks? In *Proceedings of the 25th International Conference Companion on World Wide Web*, 95–96.

Yang, L.-C.; Chou, S.-Y.; Liu, J.-Y.; Yang, Y.-H.; and Chen, Y.-A. 2017. Revisiting the problem of audio-based hit song prediction using convolutional neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 621–625. IEEE.

Zangerle, E.; Huber, R.; Vötter, M.; and Yang, Y.-H. 2019. Hit song prediction: Leveraging low-and high-level audio features. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*.