A Bayesian Approach to Understanding Music Popularity

Heather Shapiro

Advisor: Merlise Clyde

Department of Statistical Science, Duke University

heather.shapiro@duke.edu; merlise@stat.duke.edu

Abstract

The Billboard Hot 100 has been the main record chart for popular music in the American music industry since its first official release in 1958. Today, this ranking is based upon the frequency of which a song is played on the radio, streamed online, and its sales. Over time, however, the limitations of the chart have become more pronounced and record labels have tried different strategies to maximize a song's potential on the chart in order to increase sales and success. This paper intends to analyze metadata and audio analysis features from a random sample of one million popular tracks, dating back to 1922, and assess their potential on the Billboard Hot 100 list. We compare the results of Bayesian Additive Regression Trees (BART) to other decision tree methods for predictive accuracy. Through the use of such trees, we can determine the interaction and importance of different variables over time and their effects on a single's success on the Billboard chart. With such knowledge, we can assess and identify past music trends, and provide producers with the steps to create the 'perfect' commercially successful song, ultimately removing the creative artistry from music making.

Keywords: Bayesian, BART, popular music, American music industry, regression trees, sentiment analysis.

1 Introduction

Popular music is typically created to be mass distributed to a broad range of music listeners. This music is often stored and distributed in non-written form and can only be successful as a commodity in an 'industrial monetary economy'.[1] The goal of popular music is to sell as much music as possible to a large quantity of people at little cost. The definition of popular music has changed largely over time as a result of new media, which allow the mass spread of the singles. However, its form is almost always the same, including a verse, a chorus, and a bridge. Due to their repetitive nature, the verse and chorus are often the easiest portions of a song to remember and are considered as primary elements of popular music.

In order for a song to be considered popular, one must look at the dissemination of each song. Over time, music has been spread more easily to different audiences through the use of different technologies such as television, radio, and most recently, the internet. In a money making industry, it is necessary for music to have attributes that the masses will appreciate. Such success and 'conformity' have been documented by The Billboard Hot 100 charts within the American music industry since its release in 1958. Before then, Billboard Magazine published hit parades beginning in 1930.

With popular music, one of the main measures of success is how much money the single earns and how many people it reaches. Songs on the Billboard lists have a much larger audience potential, and it is seen by producers as a streamline to success. If there is a specific type of song or specific attributes of a song that garners more success, all producers would strive for this ideal song. By using a Bayesian Additive Regression Tree model, we look to see the effects of such attributes over time to see if there is a way to predict whether or not a given song will be on the Hot 100 list, its survival time on the list, and its peak rank. A sentiment analysis will also be used on song titles and lyrics to understand the importance of the written word in popular music.

2 Data

Two data sets were used for this analysis. The data for the million popular tracks comes from the Million Song Dataset, a collaborative project supported by the National Science Foundation and created by The Echo Nest, a music intelligence platform, and Columbia University's LabROSA (Laboratory for the Recognition and Organization of Speech and Audio)[2]. The dataset was created in order to encourage collaboration amongst students with regards to research on large scale commercial algorithms.

[1] Tagg, Philip (1982), Analysing Popular Music: Theory, Method and Practice, Popular Music (2): 41

[2] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

The Million Song Dataset is made up of one million popular tracks, which were selected through a five step process. First, 8,950 songs were downloaded from the most 'familiar' artists according to The Echo Nest. Approximately 15,000 songs were then found using the top 200 tags according to The Echo Nest and songs from 100 different artists with these same terms were matched. Additionally, songs and artists from the Computer Audition Lab 500-song (CAL500) data set, which consists of 500 popular music songs that were annotated by at least three listeners [3], were used. Another 475,000 songs were selected by getting songs defined as being 'extreme' according to The Echo Nest search parameters (i.e. songs with highest energy, lowest energy, tempo, song hotttnesss,etc.). Lastly, the next 500,000 songs were found using a random walk algorithm along similar artists from the 100 most familiar artists. Table 1 shows a breakdown of the statistics for the dataset provided by LabROSA.

Table 1: Statistics given by LabROSA on the Million Song Dataset.[4]

Million Song Dataset Statistics
1,000,000 songs/HDFS files
273 GB of data
44,745 unique artists
7,643 unique terms (The Echo Nest tags)
2,321 unique musicbrainz tags
43,943 artists with at least one term
2,201,916 asymmetric similarity relationships
515,576 dated tracks starting from 1922
18,196 cover songs identified

A 10,000 song subset of the Million Song Dataset was used in order to make compassion time easier for parsing and modeling. After the song files were parsed, the songs were then matched up to the Billboard dataset[5], which contains information on 39,708 singles on the Pop genre of the Hot 100 list. This includes information pertaining to how many weeks the song survived on the list and its highest rank during its duration on the list. The songs in the million song dataset that were not on the list were given values of zero and used for comparison to predict whether a song would be on the Billboard Hot 100 chart.

This large dataset will be contributed to the LabROSA efforts as an R wrapper. Previous attempts had been made to do these computations in R; however, the functions crashed due to R's functionality

Table 2: Field List Provided by the Million Song Dataset

Field name	**Description**
analysis sample rate	sample rate of the audio used
artist 7digitalid	ID from 7digital.com or -1
artist familiarity	algorithmic estimation
artist hotttnesss	algorithmic estimation
artist id	Echo Nest ID
artist latitude	latitude
artist location	location name
artist longitude	longitude
artist mbid	ID from musicbrainz.org
artist mbtags	tags from musicbrainz.org
artist mbtags count	tag counts for musicbrainz tags
artist name	artist name
artist playmeid	ID from playme.com, or -1
artist terms	Echo Nest tags
artist terms freq	Echo Nest tags freqs
artist terms weight	Echo Nest tags weight
audio md5	audio hash code
bars confidence	confidence measure
bars start	beginning of bars, usually on a beat
beats confidence	confidence measure
beats start	result of beat tracking
danceability	algorithmic estimation
duration	in seconds
end of fade in	seconds at the beginning of the song
energy	energy from listener point of view
key	key the song is in
key confidence	confidence measure
loudness	overall loudness in dB
mode	major or minor
mode confidence	confidence measure
release	album name
release 7digitalid	ID from 7digital.com or -1
sections confidence	confidence measure
sections start	largest grouping in a song, e.g. verse
segments confidence	confidence measure
segments loudness max	max dB value
segments loudness max time	time of max dB value, i.e. end of attack
segments loudness max start	dB value at onset
segments pitches	chroma feature, one value per note
segments start	musical events, note onsets
segments timbre	texture features (MFCC+PCA-like)
similar artists	Echo Nest artist IDs (sim. algo. unpublished)
song hotttnesss	algorithmic estimation
song id	Echo Nest song ID
start of fade out	time in sec
tatums confidence	confidence measure
tatums start	smallest rythmic element
tempo	estimated tempo in BPM
time signature	estimate of number of beats per bar, e.g. 4
time signature confidence	confidence measure
title	song title
track id	Echo Nest track ID
track 7digitalid	ID from 7digital.com or -1
year	song release year from MusicBrainz or 0

with different types of list objects. Table 2 details the list of variables provided by the Million Song Dataset.

2.1 Data Parsing

These song files were provided in one million different Hadoop Distributed File System (HDFS) song files, which each contain one track and all the related information (i.e. artist information, release information, audio analysis of the track, etc). The choice to use HDFS files was to minimize storage space, and to allow for people to work with subsets of the data as opposed to the dataset in its entirety. However, due to the number of directories, songs, and metadata this made the data extremely difficult to combine into one working dataset. The lab provided a subset of the songs that could be accessed through an SQLite api; however, this subset did not include all 56 variables so the HDFS files were individually parsed instead. In order to work with the data in R, a python script was written to parse each individual HDFS file and create one large CSV. The parser accommodated for the variables with different object formats (i.e., strings, numbers, or lists). Memory and time proved difficult to work with due to the number of open file limitations on different computers. Running this script in parallel only slowed down the process instead of increasing the speed due to the scripts accessing the same files at the same time.

2.2 Data Manipulation

In order to combine the two datasets, a SQL query was made to merge datasets on artists and titles that overlapped. The Billboard dataset added information on highest peak, and overall time on the charts. Due to differences in data entry in both datasets, the variables were stripped of extra spaces, punctuation, and converted to title case in order to make sure that the correct songs were matching with each other. Approximately 185 of the 10,000 song subset matched up with the Billboard Top 100 archive. The analysis was originally applied to the songs released after 1958, the official start date of the Billboard Top 100 charts. Half of the songs in the database however, do not have a year attribute. We considered songs that did not have a year to have a value of zero, and then used those with missing years as well as those with decades after the 1950s. Certain variables, such as 'artist terms', which are the terms associated with the artist according to The Echo Nest API, were provided as lists which needed to be converted to dummy variables. Other variables of type list, such as 'sections start', were reduced by finding the average distance between sections, or the mean rate for variables such as 'segment pitches'. Variables with IDs, titles, or other pieces of identifying

[3]Turnbul, Douglas. Towards Musical Query-by-Semantic-Description Using the CAL500 Data Set. University of California, San Diego, n.d. Web. 2 Dec. 2014.

[4]Million Song Dataset, official website by Thierry Bertin-Mahieux, available at: http://labrosa.ee.columbia.edu/millionsong/

[5]"Billboard Pop 100 Spreadsheet." Bullfrogspond. N.p., n.d. Web. 17 Nov. 2014.

information such as artist name were also removed from the model as the effect of those variables can be directly measured with artist familiarity and artist hotttnesss.

The difference between artist familiarity and artist hotttnesss is that 'Familiarity' corresponds to how well known in artist is whereas 'Hotttnesss' corresponds to how much buzz the artist is getting right now on the web, in music blogs, music reviews, play counts, etc.. For instance, if we look at popular artists today we can find both their artist familiarity and artist hotttnesss scores from the Echonest using its public API. Table 3, shows the list of top 10 hottest artists according the to API on March 30^{th}, 2015. If we look at artists such as Sam Smith, we see that his hotttnesss rating is almost 1, however, his familiarity score is much lower. This could be due to the fact the he is a newer artist who is slowly gaining recognition from popular songs. Artists such as the Beatles will however have a familiarity score close to 1, but a lower hotttnesss score of about .77

Table 4: Top 10 hottest artists in the Echonest API as of March 30^{th}, 2015.

Artist	**Artist Hotttnesss**	**Artist Familiarity**
Kendrick Lamar	0.99313	0.731378
Taylor Swift	0.988943	0.862272
Sam Smith	0.984216	0.68049
Ed Sheeran	0.981664	0.739728
Calvin Harris	0.978034	0.762979
Maroon 5	0.97257	0.812026
Sia	0.971183	0.773682
Ellie Goulding	0.969354	0.745824
David Guetta	0.968211	0.812422
Avicii	0.967641	0.745896
Meghan Trainor	0.965739	0.636956
Ariana Grande	0.965026	0.697816
Hozier	0.958447	0.634739
Rihanna	0.951545	0.845075
The Weeknd	0.949068	0.654817

2.3 Sentiment Analysis

Before running the models, a simple sentiment analysis was done on each of the lyric titles in the Million Song Dataset. The sentiment analysis followed the approach suggested by Jeffrey Breen with regard to Twitter sentiment analysis [6]. A simple way of calculating a sentiment score is to use the formula:

$$Score = Number\ of\ positive\ words - Number\ of\ negative\ words$$

- If Score >0, the sentence has an overall 'positive opinion'.
- If Score < 0, the sentence has an overall 'negative opinion'.

[6]Breen, Jeffrey. "R by Example: Mining Twitter for Consumer Attitudes towards Airlines." Boston Predictive Analytics MeetUp. June 2011. Jeffrey Breen Wordpress. Web. 6 Dec. 2014.

- If Score = 0, the sentence is considered to be a 'neutral opinion'.

The analysis cross references the words in each of the song titles with an opinion lexicon of both positive and negative words which were provided by Hu and Liu. [7] A sentiment score was calculated for each song title in the model. Figure 1 displays the number of songs in the sample for each sentiment score, broken up by the Billboard Top 100 indicator variable. Almost all of the songs, however, were given a neutral score of 0. This means that there were the same number of positive words as there were negative in the song title. Furthermore, in order to assess the extremity of certain song titles, the scores were broken down into 'very positive' and 'very negative' by looking at scores that were ≥ 2 or $\leq$ -2. Surprisingly, none of the 10,000 songs were given a positive score ranking. There were about 1,200 songs that did instead receive a negative score ranking. Of those, only about 64 of them had an extreme sentiment less than -1.

The scores were fitted in a logistic regression against the binary outcome of being on the Billboard Top 100 Chart or not. The sentiment score did not seem to have a statistically significant affect on the odds of a song being on the Billboard Top 100. Without any other predictors, we can not say that a positive or negative song title would hurt or better the chances of having a successful song. We did keep the variable in the model selection process to see if there could be any interactions with other variables that would prove it useful.

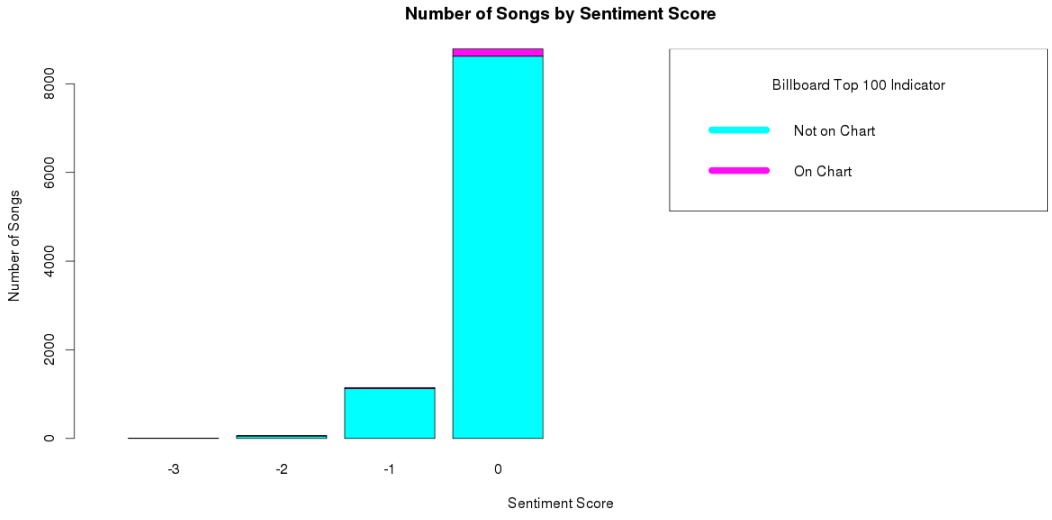

Figure 1: Number of Songs by Sentiment Score and Billboard Ranking Indicator.

[7] Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Aug 22-25, 2004, Seattle, Washington, USA,

3 Model

The data was modeled using a binary outcome with a Bernoulli Likelihood of whether or not the given song made it on to the Billboard Hot 100 list. Half of the 10,000 songs in the dataset were used for training, while the other half were used for testing purposes. First, a simple logistic regression was run with all of the variables to determine if certain variables were initially statistically insignificant in predicting Billboard success and could be removed. The fitted results were then compared to a nonparametric Bayesian regression approach using Bayesian Additive Regression Trees as well as more popular decision tree methods. Each model was rerun 100 times with different splittings of the 10,000 songs into training and testing subsets in order to asses the average predictive accuracy. In addition, due to the fact that there are so few successes in the data, our goal in model selection was to minimize the number of false positives. This means, we want to reduce the number of songs that are predicted as being successful if they in fact did not make it onto the Billboard Top 100 Chart. By reducing this number, producers who might use this method for prediction, can know with higher certainty that their song might actually be successful.

3.1 Logistic Regression

The logistic regression coefficients give the change in the log odds of being on the Billboard Top 100 Chart for a one unit increase in the predictor variable. The model was used for an initial variable selection method, in which statistically insignificant variables with high collinearity in the model were removed. Before removing any variables, the logistic regression had high multicollinearity and was predicting values close to zero or one. This inflation was removed due to the model selection and of the 80 variables that were originally provided from the parsed data, 53 variables were used for the regression trees models. A stepwise AIC was later performed on the data in order to improve accuracy in prediction, and managed to narrow down the 53 variables to only 24 variables.

3.2 Decision Trees

Tree-based methods can be useful for classification as they are simple and easy to interpret. These methods stratify or segment the predictor space, which allows us to use the mean or the mode of the training observations to make a prediction for a given observation. Tree-based methods, however, are not as successful in terms of prediction accuracy. We look to compare the accuracy of Bayesian Additive Regression Trees with more popular non Bayesian tree models, which involve producing multiple trees that are combined to yield a single prediction.

3.2.1 Bagging and Random Forest

Bagging is a more advanced classification tree method that fits many large trees to bootstrap-resampled versions of the training data, and classifies by majority vote in each subcategory. By taking the average over the different trees, the overall variance of the model is reduced, leading to improved prediction. Random Forest is a more advanced version of bagging that limits the number of variables that can be included in each tree as size m. At every tree split, a random sample of m features is drawn and included in the model. This method helps to remove the bias from important features of the model in order to assess the importance of other features.

3.2.2 Boosting

Boosting is another method used to advance classification trees. This method builds an additive model by taking the average of many trees, each of which were grown to re-weighted versions of the training data. The final classifier is found by taking the weighted average of each tree.

3.2.3 BART

Bayesian Additive Regression Tree models are not as popular as the previously mentioned decision tree methods. BART models are useful in that it creates a sum of trees model, which uses an underlying Bayesian probability model. Priors in this model strive for regularization in order to prevent any single regression tree from dominating the overall fit of the model. In the binary 'y' case, the model can be simplified to $P(Y = 1|x) = F(f(x))$, where F denotes the standard normal cdf (probit link) and f(x) is the sum of trees. In addition, the BART model was selected because regression trees are often highly praised due to their ability to flexibly fit interactions and nonlinearities.[8] Unlike Boosting and Random Forests, BART updates a set of m trees over and over, stochastic search. BART is an example of a Bayesian Markov Chain Monte Carlo (MCMC) simulation, in which we produce a draw from the posterior at each MCMC iteration. According to Chipman et al., the tree model can be explained by the following formula:

$$\pi(\theta) = \pi((T1, M1), (T2, M2), ..., (Tm, Mm), \sigma)$$

Where each (Ti , Mi) denotes a single tree. m = 200,1000,...,big,...., and T is the sum of all the corresponding μ's at each bottom node from each of the m trees plus error.

The model will help break down the different interactions of song variables over time and help to predict which attributes make up a 'perfect song'. For instance, as seen in Figure 2, the length of a song has changed significantly over time and could affect the success rate of the song. It is interesting

[8]Bleich, Adam Kapelner Justin. "BartMachine: Machine Learning with Bayesian Additive Regression Trees." (n.d.): n. pag. Web.

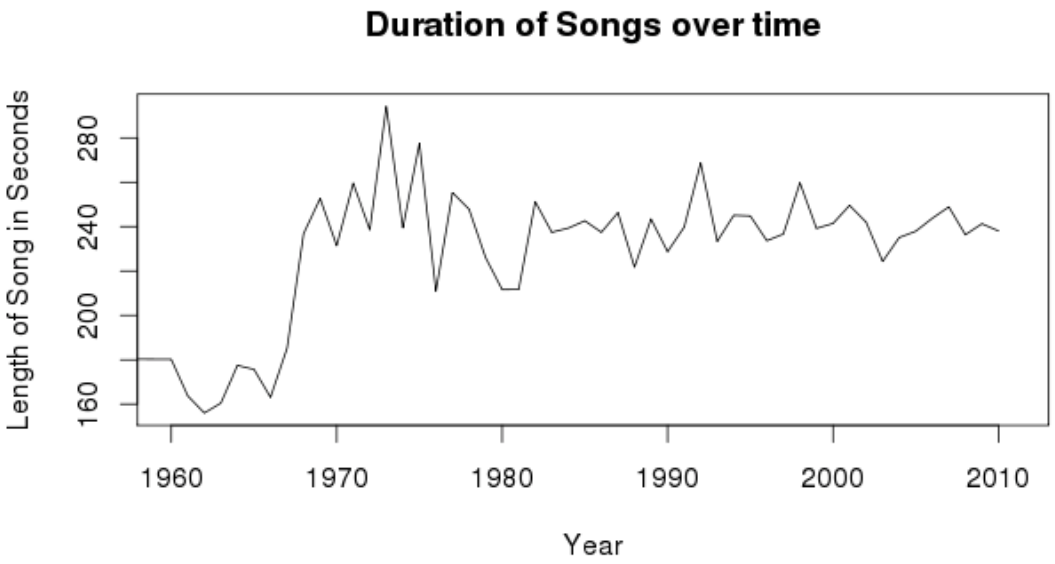

Figure 2: Duration of Songs over Time.

to note increase in the length of songs during the late 1960s, and then the peak around 1975. At this time, records were still being used to hold music, as CDs were not introduced until 1983. This seems to suggest that other factors such as radio time and commercial success, might influence the length of a song rather than technological advancements. Other users of the Million Song Dataset, considered part of the Billboard Experiment[9] have also researched popularity data from Billboard crossed with musical data from the Million Song Dataset in order to create visualizations of certain attributes of songs that might be considered commercially successful. We can see a similar trend of duration in songs in the group's findings, as well as other trends such as tempo, loudness, artist familiarity, and time signature categorized by decades as well.

Furthermore, it is apparent that this rate does not change constantly over time, so the regression tree model would create cut points in the variable as different paths down the tree. Instead of using years as a continuous variable, however, we created a new categorical variable of decades as suggested in the Billboard Experiment, which improved the accuracy of prediction in our models.

[9]Teixeira, Leonard Vilela, Letcia Lana Cherchiglia, and Rafael Almeida Batista. "Overall Visualizations." The Billboard Experiment. N.p., n.d. Web. 29 Mar. 2015.

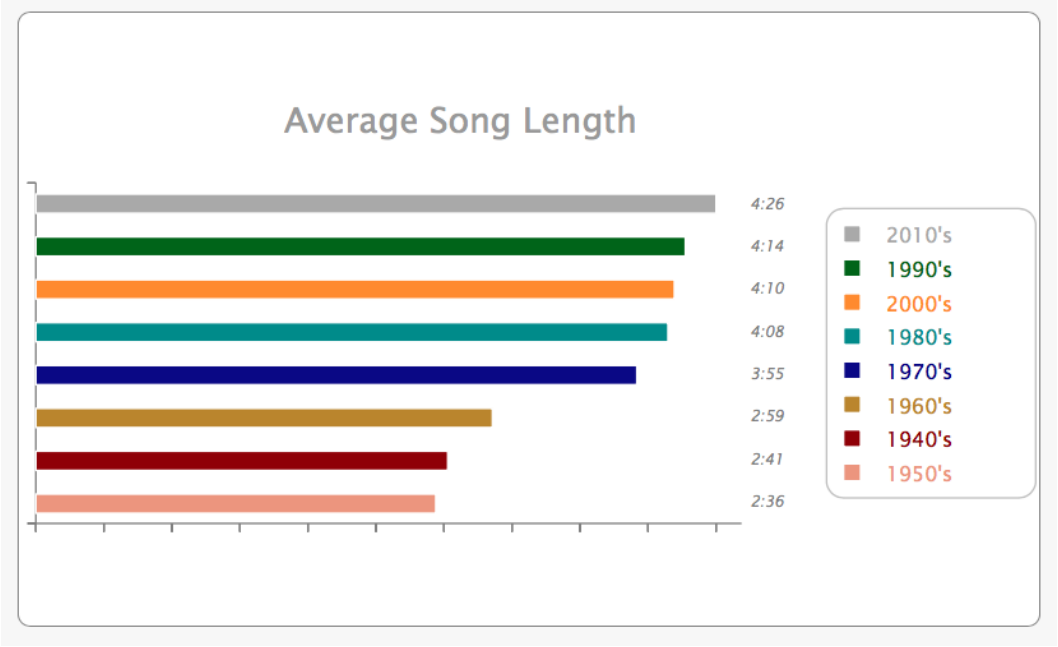

Figure 3: Duration of Songs over Time provided by The Billboard Experiment.

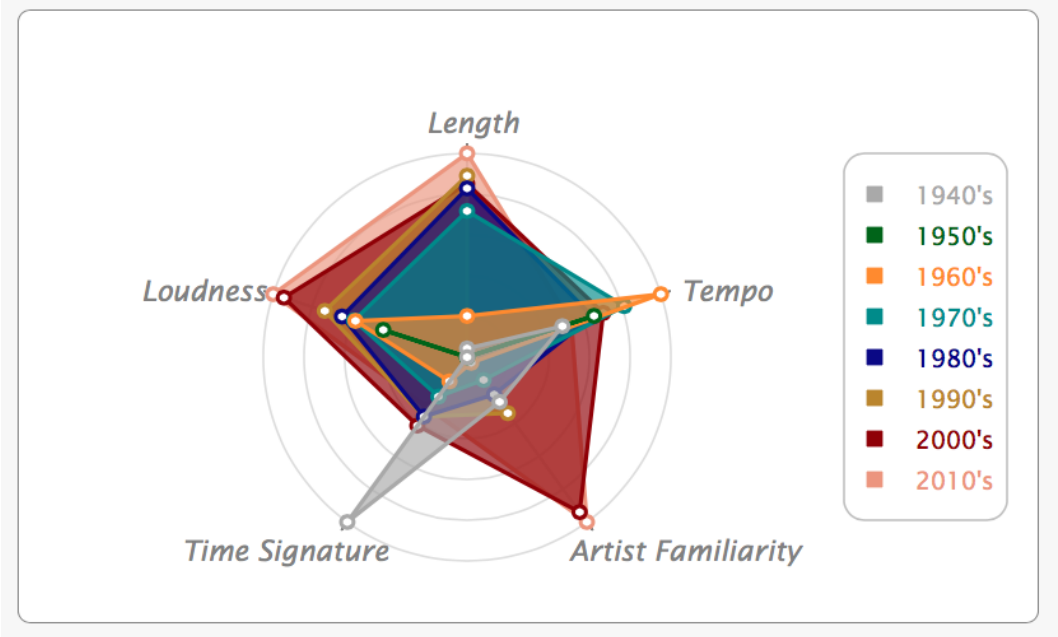

Figure 4: Radar Chart of Variables by Decades provided by The Billboard Experiment.

4 Results

4.1 Logistic Regression

The logistic regression did not take into account any interactions and looked at the effect of each individual variable on the model. One song from the dataset were removed due to having a predicted probability of zero. This song was considered an outlier to the other songs in the dataset, which would cause bias in the model. After going through variable selection and conducting a stepwise AIC on the model, the model was able to predict the test data set with an accuracy of about 96.6%. The variables with the highest influence on the model were artist familiarity and artist hotttnesss. This matches up with the importance variables produced by the various different tree methods. However, it is easier to interpret the effects of each coefficient in the model with a regression. Using the results of the stepwise model, we surprisingly see that artist familiarity was removed as an important covariate. This is unlike the results of our other tree based methods, which might indicate that without considering interaction variables, artist familiarity is not necessarily indicative a successful song.

Table 5: Results of the Logistic Regression.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-7.45	1.02	-6.68	0.00
hall1	-2.21	0.83	-2.65	0.01
usa1	-1.39	0.59	-2.34	0.02
alternative1	-1.139	0.77	-1.48	0.14
indie1	-1.02	0.55	-1.88	0.06
electronic1	-.97	0.30	-3.27	0.00
artist_mbtags_count	-.46	0.25	-1.88	0.06
start_of_fade_out	-.05	0.02	-2.51	0.01
artist_longitude	-.01	0.00	-2.33	0.02
loudness	.15	0.04	4.05	0.00
bars_start	.17	0.06	2.60	0.01
states1	.41	0.26	1.56	0.12
positive	.51	0.19	2.63	0.01
rock1	.58	0.30	1.93	0.05
hip1	.74	0.28	2.66	0.01
decades2000s	.94	0.42	2.24	0.02
decades2010s	1.22	1.19	1.03	0.30
segments_confidence	1.24	0.66	1.87	0.06
decades1990s	1.78	0.43	4.10	0.00
dance1	2.07	0.64	3.25	0.00
decades1960s	2.87	0.64	4.47	0.00
decades1980s	2.92	0.46	6.34	0.00
decades1970s	3.08	0.50	6.19	0.00
artist_hotttnesss	5.35	0.87	6.17	0.00

As seen in Table 4, artist hotttnesss has the largest affect on the log odds of whether or not a song might be on the Billboard Top 100. Without considering interactions, for every unit increase of artist hotttnesss, the log odds of a successful song increases by a factor of 5.35, indicating that

attributes about the song might not be as important. We also see that decades and certain tags from musicBrainz such as 'hip' and 'rock', also lead to increased odds of a successful song. The model successfully predicted the testing data set with an accuracy of 98.12%.

4.2 Simple Tree-Based Methods

The simple tree based method takes into account the number of positive and negative classifications in each bucket when creating the tree. The tree is grown using training data, by recursive splitting and was then pruned to an optimal size, which was evaluated by cross-validation. After pruning for the best possible tree, we chose one with a cross validation size of 7. This model does not take into account all of the variables introduced by the logistic regression, however, the top variables are the same as those seen in Table 4. The predictive accuracy of this model was also about 98.12%, proving to have a very similar accuracy to that of the logistic regression. However, in both cases, no songs on the Billboard Top 100 are predicted accurately. Due to the small sample of songs with a positive outcome, the models do a great job of predicting the songs that are true negatives.

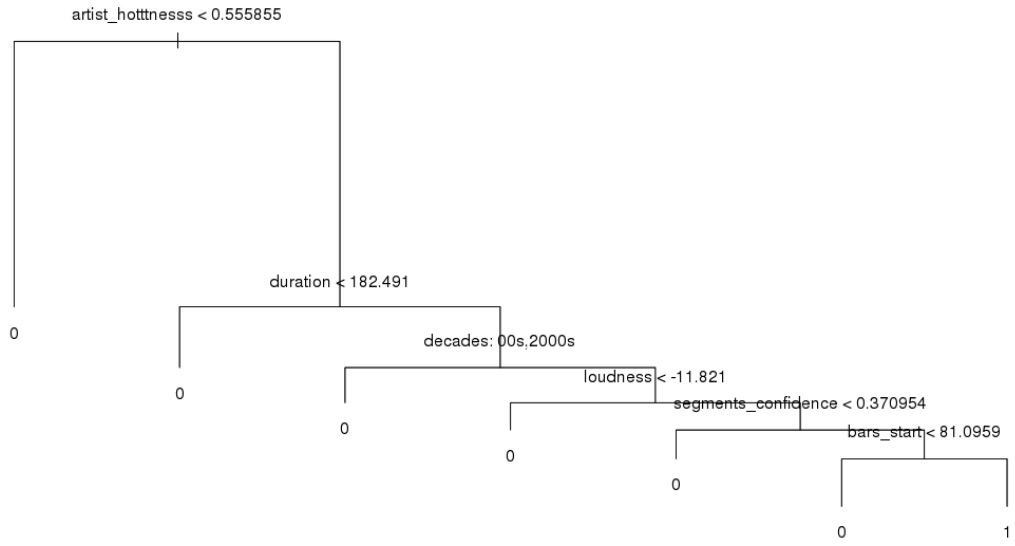

Figure 5: Tree-Based Method

4.3 BART

BART models were created for 100 different iterations of training and testing sets. On average, the predictive accuracy of the model was about 98.12%. This accuracy comes from predicting both non successful and successful songs accurately as opposed to the simple tree and logistic regression which could not predict any successes correctly. Figure 6 displays the variable importance plot for

one iteration of BART. Similarly to other models, artist hotttnesss and artist familiarity were the most important variables in the model. Interestingly, the location of an artist's hometown (or origin) was also of high importance.. This might imply that artists from different areas of the world might have different musical inspirations which lead to different types of music. If this is the case, producer might be more inclined to look for new talent in areas with larger success. In addition, the songs in the dataset that did not report a year, seem to have different features than those with reported decades.

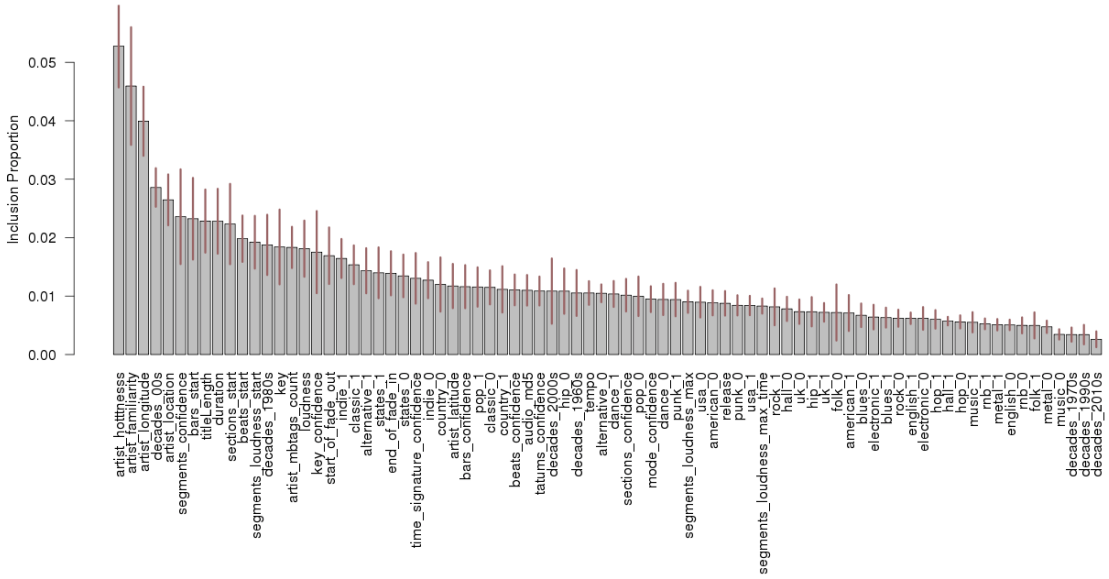

Figure 6: Variable Importance Plot for One Iteration of BART

BART also returns a set of the most important interactions between variables that contribute to the different tree models. Figure 7 shows that many of the interactions have very similar importance rankings for this iteration of BART. However, the most important interaction with the highest relative influence in this BART model is 'artist longitude x artist hotttnesss'. This would seem to imply that artists in different locations might have different chances of gaining higher hotttnesss ratings. In addition, the interaction might also be a proxy for genre, as certain locations are known to inspire artists in different genres. For instance, Atlanta is considered to be the capitol of hip-hop; artists that come from this area might have different song attributes in general from someone who is from the midwest or other geographic regions.

4.4 Bagging and Random Forest

Bagging and Random Forest methods were performed on 100 different iterations of training and testing sets with all 53 of the variables included in every tree. On average, the predictive accuracy of

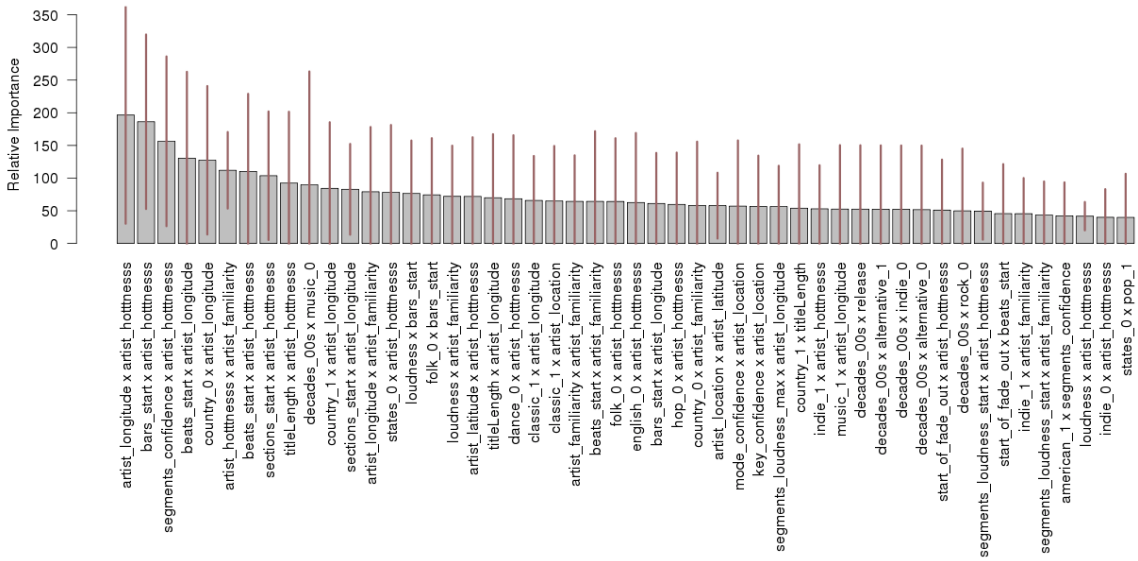

Figure 7: Relative Interaction Importance for One Iteration of BART

the Bagging was about 98.16%, whereas for Random Forests it was about 98.2%. In this instance of the training and testing sets, we see that artist hotttnesss and artist familiarity swap as being the most important between the two models. Figure 8 and Figure 9 display the importance on both the mean decrease in accuracy and mean decrease in GINI score for one instance of the Bagging and Random Forest methods. These measures are determined during the out of bag error calculation phase. The mean decrease in accuracy measures how much the accuracy of the model decreases as a result of the exclusion of a single variable. As a result, the more important variables have larger mean decreases in accuracy and are more important for classification of the data. For example, in Figure 8, we see that removing artist hotttnesss decreases the overall accuracy of the trees by a measure about 20. Mean decrease in GINI, on the other hand, is a measure of node impurity in tree based classification. Variables with higher decrease in GINI values, play a greater role in partitioning the data into the defined classes.

4.5 Boosting

Boosting was also performed on 100 different iterations of training and testing sets. On average, the predictive accuracy of the model was about 98.09%. We again see that artist hotttnesss is the lead contributor to the different models. Figure 10 measures the relative importance of each variable on the resulting model. In this model, we see more song attributes being important. For instance, while artist hotttnesss, decades, and artist familiarity have a large influence on the trees, sections

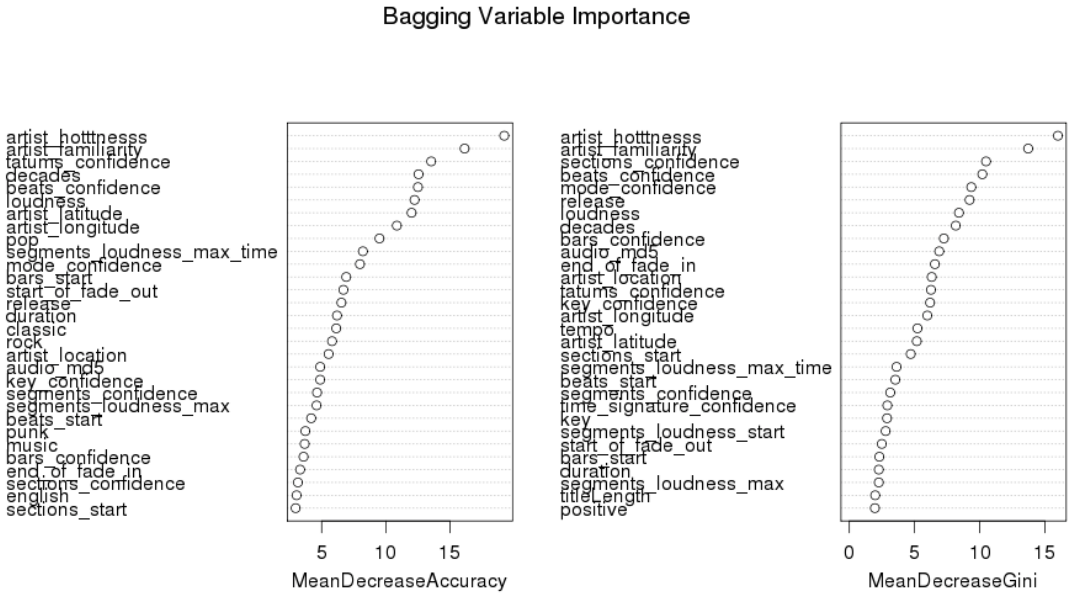

Figure 8: Variable Importance Plot for One Iteration of Bagging

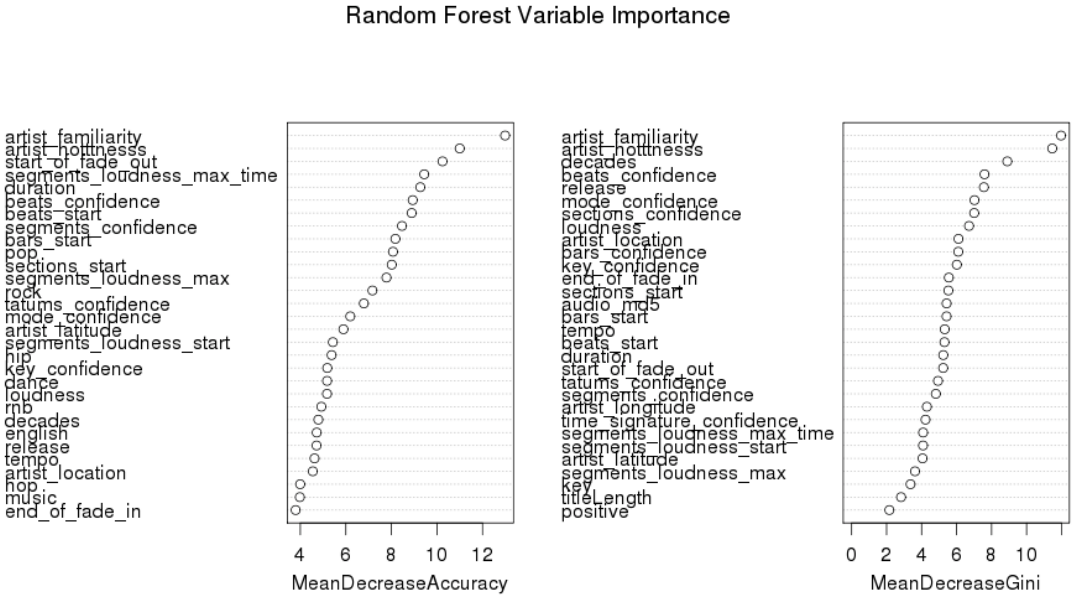

Figure 9: Variable Importance Plot for One Iteration of Random Forests of Size 7

confidence, loudness, and beats confidence have a very large effect as well. These values are also seen to be important in the other models, but are not as high up.

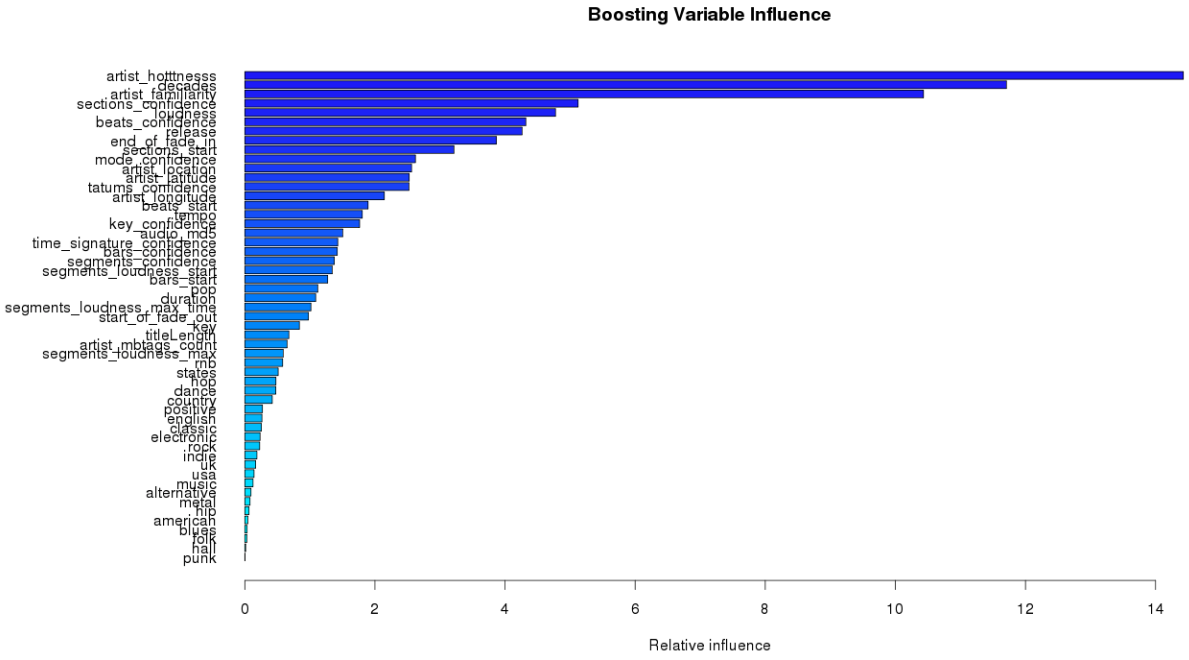

Figure 10: Variable Importance Plot for One Iteration of Boosting

5 Discussion

Based on the results, it is evident that predicting music popularity for popular songs is not an easy feat. Songs vary so vastly, and can be different for every listener that it is often difficult to quantify an aspect that is interpreted by the ear. However, through the use of intricate tree modeling, it appears the songs' attributes only go so far in helping to predict success. Artist familiarity and artist hotttnesss, are the leading contributors to whether or not a song will be successful. In addition, the results show that different decades have different underlying features, such as song duration, tempo, and loudness.

All of the models have very similar predictive accuracies as well as the same top 10 important features in slightly different orders. The logistic model and simple tree, however, did not predict any true positives correctly, which might limit their overall success. The other tree based methods, however, predicted true positives almost identically, showing that with this dataset, none of the models are necessarily better than the others. A limitation of the model, however, is that there are so few Billboard Top 100 successes. While all of the models predict with high accuracy, that is because most of the songs are being predicted as non-successes and there are very few songs

predicted as successful. Using the full million song dataset would help this limitation by providing more successes for the models to be trained on.

While our model is not a predictive model, it can be used to asses how well a song might do in the future. Future studies might include replicating the dataset for brand new artists, and seeing whether despite not having a high artist familiarity rating, a new artist's song will be on the Top 100. Recreating this dataset from Echonest proved to be difficult as most new artists do not have tags or records for the attributes that were used in this model. Future work could also include predicting the highest peak of a song that is on the Billboard list, as well as a survival analysis of how long a song might stay on the charts. If a new dataset is created, this kind of method can be useful to music producers who are trying to create a new commercial hit. For those artists with high familiarity and hotttnesss, a successful song will not be measured by the song attribute itself, but instead by the person who sings it as well as the terms it is associated with. For instance, artists with tags of indie or vocalist, but with a high hotttnesss rating, will have less of a chance of a success on the Billboard Top 100.

Acknowledgments

Thank you to my advisor, Professor Merlise Clyde, for her continuous guidance and encouragement throughout this process.

References

"Billboard Pop 100 Spreadsheet. Bullfrogspond. N.p., n.d. Web. 17 Nov. 2014.

Bleich, Adam Kapelner Justin. "BartMachine: Machine Learning with Bayesian Additive Regression Trees. (n.d.): n. pag. Web.

Breen, Jeffrey. "R by Example: Mining Twitter for Consumer Attitudes towards Airlines." Boston Predictive Analytics MeetUp. June 2011. Jeffrey Breen Wordpress. Web. 6 Dec. 2014.

HA Chipman, EI George, and RE McCullock. BART: Bayesian Additive Regression Trees. *The Annals of Applied Statistics,* 4(1):266-298, 2010.

Tagg, Philip (1982), Analysing Popular Music: Theory, Method and Practice, Popular Music (2): 41.

Teixeira, Leonard Vilela, Letcia Lana Cherchiglia, and Rafael Almeida Batista. "Overall Visualizations." The Billboard Experiment. N.p., n.d. Web. 29 Mar. 2015.

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. *The Million Song Dataset.* In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

Turnbul, Douglas. Towards Musical Query-by-Semantic-Description Using the CAL500 Data Set. University of California, San Diego, n.d. Web. 2 Dec. 2014.