Paper 11500-2016

Lyric Complexity and Song Popularity: Analysis of Lyric Composition and Relation among Billboard Top 100 Songs

Yang Gao, Management Information Systems Graduate Student, Oklahoma State University

John Harden, OSU Marketing Analytics Certification (in progress), MBA Candidate, Oklahoma State University

Vojtech Hrdinka, SAS and OSU Data Mining Certification, SAS Certified Predictive Modeler Using SAS Enterprise Miner, SAS Certified Base Programmer, SAS Certified Business Analyst, MBA Graduate, Oklahoma State University

Chris Linn, SAS and OSU Marketing Analytics Certification, SAS Certified Predictive Modeler Using SAS Enterprise Miner, SAS Certified Base Programmer, MBA Graduate, Oklahoma State University

ABSTRACT

Does lyric complexity impact song popularity, and can analysis of the Billboard Top 100 from 1955-2015 be used to evaluate this hypothesis? The music industry has undergone a dramatic change. New technologies enable everyone's voice to be heard and has created new avenues for musicians to share their music. These technologies are destroying entry barriers, resulting in an exponential increase in competition in the music industry. For this reason, optimization that enables musicians and record labels to recognize opportunities to release songs with a likelihood of high popularity is critical. The purpose of this study is to determine whether the complexity of song lyrics impacts popularity and to provide guidance and useful information toward business opportunities for musicians and record labels. One such opportunity is the optimization of advertisement budgets for songs that have the greatest chance for success, at the appropriate time and for the appropriate audience. Our data has been extracted from open-source hosts, loaded into a comprehensive, consolidated data set, and cleaned and transformed using SAS® as the primary tool for analysis. Currently our data set consists of 334,784 Billboard Top 100 observations, with 26,869 unique songs. The type of analyses used includes: complexity-popularity relationship, popularity churn (how often Billboard Top 100 resets), top five complexity-popularity relationships, longevity-complexity relationship, popularity prediction, and sentiment analysis on trends over time. We have defined complexity as the number of words in a song, unique word inclusion (compared to other songs), and repetition of each word in a song. The Billboard Top 100 represents an optimal source of data as all songs on the list have some measure of objective popularity, which enables lyric comparison analysis among chart position.

INTRODUCTION

The purpose of our project was to determine whether the complexity of lyrics impacted the popularity of any given song, via analysis of lyrical composition of songs on the Billboard Top 100 weekly charts from 1955 to 2015. We chose this study due to the evolving music landscape that allows listeners of music greater control over their media consumption.

We hypothesized songs with low levels of lyric complexity would increase song popularity, and would allow songs to increase their Billboard Top 100 chart position, along with improving their chart longevity.

Through analysis of open-source data, we were able to determine that lyrical complexity does impact popularity, but other factors (such as artist popularity and marketing investment) are most likely more influential on chart position (a measure of objective popularity) than our definition of lyric complexity.

The dataset compiled for this project represents a foundation for additional research. For instance, after determining the weak relationship between lyrics and chart position in our existing models, we used the data to find a much more meaningful relationship between where a song enters the charts and where it will peak. While technically outside the scope of our original project, these findings can still have value for

our identified benefactors. Additional modeling of the data (with or without additional variables) may uncover subsequent relationships of value.

A major goal of this paper is to introduce the project and dataset to interested parties so that they may continue research into this topic. A secondary goal is to demonstrate our data collection and consolidation methods for data analysts interested in compiling open-source data for their own projects. A third goal of this paper is to find interesting relationships among variables in this data set.

PROJECT CONSIDERATIONS

IDENTIFICATION OF POTENTIAL BENEFACTORS

The benefactors of this study could include musicians, record labels, song writers, and agencies representing artists, and advertisers.

This study can potentially assist musicians and record labels in marketing and advertising songs that have the greatest chance for success. The benefactors would be able to maximize their advertisement budgets for songs that have the greatest likelihood of wide popularity, thus improving their Return on Investment for those songs. Our current model is valid only for songs that have already entered the Billboard Top 100, but could provide additional insights to potential benefactors prior to releasing a song.

Further evaluation of the data set may determine relationships that could:

- Help evaluate musicians' existing but unreleased songs for a recommendation on which songs to release as singles as well as possible release dates.
- Provide recommendations to musicians and record labels on whether to alter their lyric complexity for the purpose of increasing popularity and capturing a larger audience.
- Allow commercial advertisers to use this study to determine which yet-to-be-released songs have a high likelihood for top popularity with the hopes of increasing recall of advertisements.

CONSTRAINTS AND LIMITATIONS

All songs on the Billboard Top 100 list represent a population of "popular" music. Our initial concern was that the models we expected to create may not have been applicable to songs that have never appeared on the list. Our model efforts were constrained to songs that have already achieved a measure of objective popularity (indicated by appearance on the Billboard Top 100 charts). As Entry Position was the greatest indicator of Peak Position and Longevity in our chosen model, the original concern is not valid, as the chosen model is specific to songs that have already appeared on the list.

Our ability to model popularity was constrained by the use of lyric and song metadata, and did not take into account other factors, such as Artist Popularity or Marketing Activities/Investment, that would likely explain a much higher percent of modeling variance if they were quantifiable and accessible for inclusion in the model. Artist popularity may have impacted the scope of release and popularity of songs in a way that lesser-known artists' songs would not have been impacted, and it is assumed that all songs on this list have had some level of marketing investment applied to increase their exposure, which, in turn, would impact popularity (however, the degree to which investments were made is unknown, so we could not assume a uniform investment for the songs on the list).

The population data set was not used for this analysis. Imputation of variables for songs with incomplete data were used in the sample data set, in order to include them in analysis. Imputed values were derived from a decision tree application to provide values for incomplete instances of variables Duration, Words (the count of words in a song), and RepeatedWords.

FEASIBILITY ASSESSMENT

Technical

The biggest technical hurdle we faced was the acquisition and consolidation of supplemental song data (such as duration, genre, release date, release location, etc.). Extensive effort was taken to collect supplemental data via Import.IO, but results were inconsistent.

The team then debated seeking outside support to collect the supplemental data, and identified three possible access points for the desired material: a Musicbrainz.org moderator, a Million Song Database administrator, and a local virtual-server expert. Prior to making outside contact and after ~30 hours of total research in to accessing the databases ourselves, a team member discovered the existence of and accessed the Whitburn Project dataset, which contained the majority of the desired supplemental data, and reduced this technical hurdle to a non-issue.

Operational

Validation of findings throughout the project (such as mitigating collinearity, utilizing more accurate imputation methods, and creating dozens of model iterations to determine an optimal solution given the data set) required additional effort, but ultimately led to our chosen model.

DATA COLLECTION, CLEANING AND CONSOLIDATION

All data was extracted from open-source hosts, loaded into a comprehensive, consolidated data set, and cleaned and transformed using SAS as the primary tool for analysis.

SAMPLE AND POPULATION

The initial dataset consisted of 334,484 observations, which was comprised of the songs and positions on each weekly the Billboard Top 100 list for the time period between 1955 and 2015. Within those observations there were 31,971 unique songs – representing the population – of which we were able to collect lyric and metadata on 18,102 for our sample data set. When creating samples we used 60% of the observations for training data and 40% of the observations for validation data.

DATA ACCESS

A list of collected data types and sources, and the capture and collection methods are as follows:

Billboard Top 100 data from 1955-2015

We were able to make contact with a private music enthusiast who had access to a CSV file that contained all Billboard Top 100 weekly data from 07/20/1940 to 10/24/2015. However, we found out that the records before 11/02/1955 were incomplete (often only Top 30, Top 15, etc.). To maintain the integrity of our analysis, we chose to remove (data reduction) everything before November 1955, so that we could work with full Top 100 charts for every single week between 1955 and 2015.

The variables contained within this source included Song Title, Artist, Week Date, This Week Position, and Last Week Position.

Lyric Collection

Lyric data was collected via use of the Import.Io webcrawler, from several open-source lyric websites.

The first approach taken was to use the webcrawler on websites with predictable URL formats, such as "[website].com/[song name]_[artist].htm". First, the artist and song data in the previously collected Excel spreadsheet columns were concatenated into a single cell. Next, a website address, such as www.lyricsfreak.com, was then concatenated into the column. The webcrawler was loaded with the bulk concatenations and trained to collect the lyric data, artist names and song names. This approach generated approximately 30% of the total lyric data collected.

The final approach we undertook to collect lyric data required a dual-webcrawler strategy. First, lyric websites with search tags in their URLs (such as "[website].com/search=") were concatenated with the song title and artist. Next, the webcrawler was trained with the single concatenated value to capture the song URL generated by the search and then saved. After confirming this method captured a link to the desired song page, the webcrawler was loaded with the concatenated values via the Bulk query option. Once the crawler completed gathering the links to the desired song page, a second webcrawler was built.

The second webcrawler was trained to capture the lyric data, song title and artist name from the previously-generated webcrawler links. Useful data was saved and broken links or invalid results were

deleted, and the process then was completed using other lyric websites until a sufficient body of data was collected. This method accounted for the remaining lyric data.

Supplemental Data – Whitburn Project

While some Genre and Duration data was collected along with the lyric data collections, the vast majority of supplemental data was collected with the discovery of a Billboard Top 100 consolidated dataset that contained Duration and Genre for a large percentage of the data (~93% and ~84%, respectively).

While researching the Billboard Top 100, a consolidated dataset of popular music from 1890 to 2009 was discovered, called the Whitburn Project. Coincidentally, this dataset of popular music used the Billboard Top 100 after its creation in 1955, which allowed us to directly merge the file with our existing data.

DATA CLEANING

The data cleaning processes applied to the Lyrics, Billboard Top 100 and Whitburn Project datasets removed unnecessary special characters, separated the primary artist from the featuring artists, and converted necessary special characters or word abbreviations that had the same meaning into a standardized character or word. In a very small number of entries, necessary data was not included; due to the size of the dataset, we determined the exclusion of these data points would not significantly impact the dataset as a whole.

Unique Song ID Creation

All data sources contained Artist Name and Song Title information along with their additional variable information. Knowing this, we created a unique song ID using truncated Artist and Song Title information, which would allow us to merge different datasets with the unique song ID being a primary key for merging.

After the collected data was cleaned of special characters, common expressions (such as "the"), and standardized, we created a column in each data set where the first four characters of Artist were merged with the first ten characters of the Song Title. For Example, a row with Artist Name "The Police" and Song Title "Every Breath You Take" resulted in a unique song ID "polieverybreat". The sequential steps we took are below.

1. We first removed Semicolons from the csv file (in notepad++)

We received a comma delimited CSV file for the Billboard Top 100 data. However, some of the song and artist names contained semicolons (as can be seen below). We removed these in order to preserve the structure of the document and import of variables into appropriate columns. **Error! Reference source not found.** is a screen capture of the comma-delimited CSV file for the Billboard Top 100 data.

Find result - 8 hits

Find result - 8 hits	×
Search ";" (8 hits in 1 file)	_
C:\Users\Vojtech\Dropbox\OSU\Fall 2015\MSIS 5633\BITT Project\CSV Dataset\bbsp_puvodni.txt (8
Line 22425: 09/28/1957 100 "" "JOHNNIE AND JOE" "Over The Mountains; Across The Sea"	
Line 94192: 12/14/1968 104 "" "NINA SIMONE" "Ain't Got No; I Got Life"	
Line 94323: 12/21/1968 103 "104" "NINA SIMONE" "Ain't Got No; I Got Life"	
Line 94456: 12/28/1968 102 "103" "NINA SIMONE" "Ain't Got No; I Got Life"	
Line 94574: 01/04/1969 99 "102" "NINA SIMONE" "Ain't Got No; I Got Life"	
Line 94700: 01/11/1969 96 "99" "NINA SIMONE" "Ain't Got No; I Got Life"	
Line 94817: 01/18/1969 95 "96" "NINA SIMONE" "Ain't Got No; I Got Life"	
Line 94930: 01/25/1969 94 "95" "NINA SIMONE" "Ain't Got No; I Got Life"	
Esearch ";" (8 hits in 1 file)	

Display 1. Comma-delimited CSV file showing Billboard Top 100 data.

2. We then filtered out records before 2 November 1955

We only used data from 11/2/1955 through 10/3/2015. The records from before November 1955 were incomplete and were manually deleted from the CSV file.

3. Next, we cleaned Artist Name and Song Title

The first step to creating the unique identifier of a song was to remove special characters or word abbreviations. This was done using the following SAS code:

```
/*creates variables ArtistName and SongName with the length of 60*/
data dataset;
  set library.dataset;
   length ArtistName $60 SongName $60;
      SongName = lowcase(XSongName);
/*puts the variable in lower case*/
    _ArtistName = lowcase (XArtist);
run;
/*manipulating ArtistName and SongName*/
data dataset;
  set dataset;
    _SongName = tranwrd (_SongName,'-',' '); /*replace dash with space*/
    _SongName = tranwrd (_SongName, '$', 's');
    _SongName = tranwrd (_SongName,'&','and');
    ArtistName = tranwrd ( ArtistNAme, 'feat.', 'featuring');
    ArtistName = compress( ArtistName, '/?()""''@#%:.,!*');
/*remove these characters and remaining asterisks*/
run;
```

Additionally, we removed all instances of the preposition "the" from the Artist Name and Song Title. We did this to address the fact that different sources recorded artist and song names differently; for example, "The Police" band have been found to be recorded as both "Police" and "The Police". By removing the preposition, we prevented "The Police – Every Breath You Take" and "Police – Every Breath You Take" from creating two unique ID's in the next step. In addition, the Artist Name entries were stripped from featuring artists. The SAS code to strip featured artists we used was:

```
/*Create Artist and FeaturingArtist from _ArtistName; use 'featuring' as
delimeter*/
data dataset (rename=(_ArtistClean=_ArtistName));
  set dataset;
    _ArtistName = tranwrd(_ArtistName,'featuring','#');
    _ArtistClean = strip(scan(_ArtistName,1,'#'));
    _Featuring = strip(scan(_ArtistName,2,'#'));
drop _ArtistName;
run;
```

4. Create the UniqueID variable

In the final step of creating UniqueID variable, we concatenated the first four characters from the cleaned Artist Name with the first ten characters from the cleaned version of Song Title. The SAS code used was:

```
data library.dataset_clean (drop=_featuring);
  set dataset;
    length UniqueID $15;
    NoblArt = compress (_ArtistName,' ');
    NoblSong = compress (_SongName,' ');
    art5 = substr(NoblArt,1,4);
    song10= substr(NoblSong,1,10);
    UniqueID = cats(art5,song10);
drop art5 song10 NoblArt NoblSong;
run;
```

Lyrics Cleaning

The biggest concerns with the lyrics dataset was the inconsistency of songs having the entire lyrics written out, and songs that displayed the chorus only once and wrote the word chorus in place of the entire chorus. To eliminate this inconsistency we decided to remove all lyrics that contained the word "chorus" by using find and replace of *chorus* within excel. We searched all missing lyrics again through a webcrawler in attempts to gain lyrics for songs that had empty lyric cells. We conducted other cleaning through Find and Replace. The following are examples of additional information removed from lyrics data:

- Submitted on* (This occurred at the end of many songs)
- Submitted by* (This occurred at the end of many songs)
- Written by* (This occurred at the end of many songs)
- Unfortunately we were* (This occurred when lyrics weren't available from www.letssingit.com)
- (*),[*],{,}(Many songs had artist name, instrumental, or verse within parenthesis and brackets)

We also needed to find particular words and see if they were officially part of the song or information about the song. These words were "copyright" and "instrumental". We used Find in excel and looked at these words on a case-by-case basis (Note: sorting the dataset alphabetically significantly assisted all manual cleaning efforts, as we created a numbered column of the data prior to sorting so that it could be easily resorted back to the original).

DATA TRASFORMATION

We utilized the Max Normal Transformation function in SAS Enterprise Minor to assist in deciding if transformation is necessary, but nearly all of our variables were within normal parameters. While there were a couple of instances of slight skewness, we thought a transformation would make any results calculated with the transformed variable too difficult to explain in actionable terms for our benefactors.

DATA REDUCTION

Our collection for the Billboard Top 100 data from 1955-2015 required removal of instances that occurred before the scope period (IE. <1955). Import.IO collected URLs for lyrics, but were not necessary to include in our dataset; they were removed as well.

Within the Whitburn Project data set, duplicate variables and unnecessary metadata (such as Production Label) were also deleted.

Prior to running the models, correlation and collinearity tests were performed to omit variables that would skew results if included together.

DATA CONSOLIDATION

Our data was collected from multiple sources which we had to merge. To create a table of individual songs appearing in the Billboard Top 100 chart between 1955 and 2015, the table of Billboard Top 100 weekly records with 334,474 rows was aggregated using Unique ID as a unique identifier of a song. The aggregated table contained 28,002 unique songs. The merging was done using the "join tables" command in SAS Enterprise Guide Query builder, with Unique ID as a primary key. In the first phase, lyrics and other variables, e.g. duration and genre, from various sources were merged to the Billboard Top 100 Unique Songs table using Left Joint. The diagram below shows the connection.

Display 2 is a sample display of the "Join Tables" command in SAS Enterprise Guide Query Builder, using the UniqueID variable for the merger.



Display 2. UniqueID variable used to join data sets

FINAL DATA SET CHARACTERISTICS AND NEW VARIABLES

The final dataset contained the following percentages of our key variables:

- Total Unique Songs: 28,002
- Missing lyrics: 9,647 (34.5%)
- Missing duration: 1,812 (6.5%)
- Missing genre: 4,447 (15.8%)
- Missing BPM: 7,471 (26.7%)

We then created several variables from the aggregated data set.

Variable Name	Variable - Full Name	Variable Meaning
AvgTWPos	Average This Week Position	Average position of the song.
AvgTWbins	Average This Week Position Bined	AvgTWPos with bins of 10 spots.
BestTWPos	Best This Week Position	Best position the song has ever achieved.
BestTWPbins	Best This Week Position binned	BestTWPos with bins of 10 spots.
WeeksInChart	Weeks in Chart	Number of weeks the song has stayed in the Billboard Top 100 Chart.
Month		Month when the song first entered the chart.
Year		Year when the song first entered the chart.
Тор50		The song has been in the top 50
Тор40		The song has been in the top 40
Тор30		The song has been in the top 30
Тор20		The song has been in the top 20
Тор10		The song has been in the top 10
Тор5		The song has been in the top 5
WorstTWPos	Worst This Week Position	What was the lowest position of the song during its presence in the chart
EntryDate		Exact date when the song entered the chart.
ExitDate		Exact date when the song left the chart.

Table 1 shows additional variables we created after the aggregation process.

Variable Name	Variable - Full Name	Variable Meaning
AvgTWPos	Average This Week Position	Average position of the song.
AvgTWbins	Average This Week Position Bined	AvgTWPos with bins of 10 spots.
BestTWPos	Best This Week Position	Best position the song has ever achieved.
BestTWPbins	Best This Week Position binned	BestTWPos with bins of 10 spots.
WeeksInChart	Weeks in Chart	Number of weeks the song has stayed in the Billboard Top 100 Chart.
Month		Month when the song first entered the chart.
Year		Year when the song first entered the chart.
Тор50		The song has been in the top 50
Тор40		The song has been in the top 40
Тор30		The song has been in the top 30
Тор20		The song has been in the top 20
Тор10		The song has been in the top 10
Тор5		The song has been in the top 5
WorstTWPos	Worst This Week Position	What was the lowest position of the song during its presence in the chart
EntryDate		Exact date when the song entered the chart.
ExitDate		Exact date when the song left the chart.

Table 1. Created Variables after Aggregation

We then created additional variables we would need for our models. Two example variables we created and the associated SAS code used to create them are:

• Variable "Words" based on the word count of lyrics:

```
data dataset;
   set dataset;
   words = countw(_lyrics);
run;
```

• Variable "WPM" (Words Per Minute):

```
data dataset;
   set dataset;
   wpm = words/duration_decimal;
run;
```

MODELING TECHNIQUES

In terms of modeling techniques it is important to monitor for collinearity among input variables. By excluding those variables which had a VIF over 10, we ensured a more accurate model. Another step is to determine which variables may be strongly related to the target variable. We found, of the input variables, the most strongly related to be Entry Date, Duration Decimal, Repeated Ratio, Repeated Frequency, WPB, WPM, and Words.

Combining variable importance and VIF, we used five input variables in many of our initial models. These variables were Duration Decimal, Repeated Ratio, Repeated Frequency, WPB, and WPM.

To build models, we created six different target variables. Five of the target variables were continuous/ordinal, so we used linear regression for prediction modeling. One variable was binary, so we used decision trees for classification and logistic regression, neural network, and auto neural network for prediction.

The variables used for linear regression were: BestTWpos, BestTWposBINS (in sections of 10 positions – IE 1-10, 11-20), WeeksinChart, PK, DaystoPK. We chose Linear Regression for these variables because they were continuous, and we hoped to find a linear relationship of the input variables that measured song simplicity to chart success. When conducting linear regression we examined each combination of variables with Stepwise, Forward, Backward, and Ensemble selection method. These were used to see if different variables were accepted in the models, along with a hope of improving the variance explained by the model.

The variable used for decision trees, logistic regression, and neural networks was Top40 (1 if it peaked in Top 40, 0 if not), These models were used because the target variable was binary along with ease of interpretation of decision trees, and we found it important to be able to see if input variables that measured song simplicity allowed us to see if a song would peak in the Top 40 regardless of position within the Top 40. We chose to see Top 40 because radio charts, such as the American Top 40, would use the Top 40 songs of Billboard and the radio play would be magnified for these songs.

BUILDING THE MODELS

Due to much of our failure in creating a model that accounted for a large degree of variance explained, we tried 80+ models. These models were created through experimentation. We adjusted with filtering before and after data splitting. We found that it was better to filter before data splitting to avoid many of the observations from being in one of the subsets. We experimented with a wide arrange of variables including words, wpm, bpm, wpb, repeated words, repeated ratio, repeated frequency, entry date, text clusters.

As an example of one of our failed models, as a demonstration of the iterative nature of the project, was our initial linear regression model. The linear regression model was not able determine any independent variables significant enough to predict PK. Obviously, it was important to change modelling method and obtain, calculate and create more variables to create a model that was significant that included at least one input variable.

After many failures, we continued to experiment with differing target variables, input variables and models. All these target variables had one thing in mind, to see if we could determine chart success based on song simplicity. We began to find better success with the following models:

- 1. Days to PK Backward Regression
- 2. BestTWPos Linear Regression
- 3. Top 40 Decision Tree
- 4. Top 40 Logistic Regression

ASSESSING THE MODELS

DAYS TO PK BACKWARD REGRESSION

This model explained 6.4% of variance and used the significant variables of DurationDecimal, Repeated Frequency, and Repeated Ratio. Even though adjusted r-square figure is only 6.4%, the overall model is significant, and is one of the best models. This model produces an equation of: Days to $PK = 1.2447 + .2207^*(DurationDecimal) + .6409^*(RepeatedFrequency) - 3.1748^*(RepeatedRatio).$

Output 1 shows the Analysis of Variance and Model Fit Statistics for this model.

		1	malysis of	Variance			
			Sum of				
Source	I	F	Squares	Mean S	iquare F	Value H	Pr > F
Model		3	40.653417	13.5	51139	12.96 <	.0001
Error	51	.7	540.594183	1.0	45637		
Corrected To	otal 52	20	581.247601				
	Model Fit	Statis	stics				
R-Square	0.0699	Adj	R-Sq	0.0645			
AIC	27.2347	BIC	:	29.2965			
SBC	44.2577	C (p))	4.0000			
	Analysi	s of N	faximum Like	lihood Esti	mates		
				Standard			
Parameter		DF	Estimate	Error	t Value	Pr > 1	51
Intercept		1	1.2447	0.7100	1.75	0.080	02
IMP_Duration	Decimal	1	0.2207	0.0706	3.12	0.003	19
IMP_Repeated	lFrequency	1	0.6409	0.1731	3.70	0.000	02
IMP_Repeated	Ratio	1	-3.1748	1.5220	-2.09	0.033	75

Output 1. Days to PK Backward Regression Output

BESTTWPOS LINEAR REGRESSION

After recreating the original, failed model with BestTWPos target variable, we achieved an improved (compared to the original model BestTwPos model) adjusted r-square of .0366. To get this improved model, we used durationdecimal and repeated ratio as input variables. The analysis of variance SAS Enterprise Miner output (below) shows that the overall model is significant. This model produces a prediction equation of: BestTWPos = $102.5 - 2.4423^{\circ}$ (DurationDecimal) - 63.8410° (RepeatedRatio). shows the Analysis of Variance and Model Fit Statistics for this model.

			Analysis of N	Variance		
			Sum of			
Source		DF	Squares	Mean Sq	uare FV	Value Pr > F
Model		2	56275	2	8137 3	33.35 <.0001
Error		1699	1433639	843.81	3291	
Corrected	Total	1701	1489914			
	Model F	it Stati	stics			
R-Square	0.037	B Ad	j R-Sq	0.0366		
AIC	11470.956	3 BI	C 114	172.9687		
SBC	11487.275	D C(p)	2.5120		
	Analy	ysis of	Maximum Likel	lihood Estim	ates	
				Standard		
Parameter		DF	Estimate	Error	t Value	Pr > t
Intercept		1	102.5	6.6254	15.47	<.0001
IMP_Durati	onDecimal	1	-2.4423	1.0492	-2.33	0.0200
IMP_Repeat	edRatio	1	-63.8410	8.3252	-7.67	<.0001

Output 2. BestTWPOS Linear Regression Output

TOP 40 DECISION TREE

Decision Tree was used as classification, not prediction, and we ended up with a Validation Misclassification rate of 37%. While the general consensus is that a reasonable classification model has a Validation misclassification rate below 30%, this model was the best one we were able to create. While the model is poor at identifying songs that will reach the top 40, it is good in determining which songs will not reach the top 40. (see the specificity and sensitivity values below). First, we built a "maximal" decision tree which was too complex and did not classify well on the validation dataset. Therefore, we have decided to prune it back to achieve a tree structure that can be seen below. This is easier to follow and make conclusions from, and it is not as complex as the maximal tree. Two Decision Tree metrics of note, derived from Output 3, are:

- Validation 17.88% Sensitivity (True Positive Rate; poor at identifying songs that will reach top 40).
- Validation 92.95% Specificity (True Negative Rate; great at identifying songs that won't reach top 40).

Event Clas	sification T	able							
Data Role=	TRAIN Target	=Top40 Targe	t Label=' '						
False	True	False	True						
Negative	Negative	Positive	Positive						
508	972	80	142						
Data Role=VALIDATE Target=Top40 Target Label=' '									
Felce	True	Felse	True						
False	True	False	True						
False Negative		False Positive							

Output 3. Top 40 Decision Tree Event Classification Table

Observing the decision tree (Output 4), we saw that Repeated Ratio was the most important variable, and therefore the criteria for the first split. If a song has a repeated ratio of .7405 or less, it is likely to be classified as not making the Top 40 (62.91%). If it is greater than or equal to .7405, the tree needs to be split further. Repeated Freuqency is the second split, and of those songs that had a repeated ratio greater than or equal to .7405, if a song has a repeated frequency less than 2.2989, then 100% are classified as reaching the Top 40. The third and final split is based from Duration Decimal.



Output 4. Top 40 Decision Tree Output

TOP 40 LOGISTIC REGRESSION

Our validation misclassification rate was 39%. While the general consensus is that a reasonable classification model has a Validation Misclassification Rate below 30%, this model was the second best we were able to create. Other notable findings with this model, as can be seen in Output 5:

- As Duration Decimal increases 1 unit, the odds of song peaking in top 40 increases 1.253%
- As RepeatedRatio increases 1 unit, the odds of song peaking in top 40 increases 80.39%

-2 Log Likel:	ihood	Like	lihood				
		6	Ratio				
Only	Covariate	s Chi-	Square	DF Pr	: > ChiSq		
2263.620	2202.32	:0 6	1.3005	2	<.0001		
		Analysis o	f Maximum	Likelihood E	Stimates		
			Standard	Wald		Standardized	
Parameter	DF	Estimate	Error	Chi-Square	Pr > ChiSq	Estimate	Exp(Est)
Intercept	1	-4.2512	0.5147	68.23	<.0001		0.014
IMP_DurationDecim	al l	0.2256	0.0766	8.67	0.0032	0.0837	1.253
IMP_RepeatedRatio	1	4.3869	0.6395	47.05	<.0001	0.2050	80.392
Odds Rat:	io Estima	ites					
		Poi	nt				
Effect		Estima	te				
IMP_DurationDecim	al	1.2	53				
IMP RepeatedRatio		80.3	92				

Output 5. Top 40 Logistic Regression model output.

PROJECT PIVOT

We still were not achieving desired success for many of our models. We adjusted our focus to see if we could predict where a song will peak on the charts, since we knew what position a song entered the chart. This required the use of the input variable Entry Position. Entry Position had the highest variable worth of all input variables, so it was likely going to make more of an impact. A perfect example is listed below featuring BestTWPos as the Target variable. The new models used variable selection criteria of Backward and Forward.

BACKWARD WITH ENTRY POSITION

The Backward model was significant, containing an adjusted r-square of .2212. The significant variables were EntryPos and DurationDecimal. This backward model produced a prediction equation of BestTWPos = $57.9927 + .5470^{\circ}(EntryPos) - 4.3111^{\circ}(DurationDecimal)$. Output 6 shows these figures.

			Analysis of	Variance			
			Sum of	:			
Source		DF	Squares	: Mean	Square	F Value	Pr > F
Model		4	104613)	26153	37.93	<.0001
Error		516	355762	689.	460658		
Corrected T	Total	520	460375	5			
	Model Fi	t Stat	istics				
R-Square	0.2272	A	dj R-Sq	0.2212			
AIC	3410.1848	E	IC 3	412.2815			
SBC	3431.4635	C	(p)	5.0000			
	Analy	sis of	Maximum Like	lihood Est	imates		
				Standard	L		
Parameter		DF	Estimate	Error	t Val	ue Pr	> t
Intercept		1	57.9927	18.5893	3.	12 0	.0019
EntryPos		1	0.5470	0.0481	11.	38 <	.0001
IMP_Duratio	nDecimal	1	-4.3111	1.8134	-2.	38 0	.0178
IMP_Repeate	dFrequency	1	2.1496	4.4510	0.	48 0	.6293
IMP_Repeate	dRatio	1	-59.4313	39.0905	-1.	52 0	.1290

Output 6. Backward with Entry Position model output.

FORWARD WITH ENTRY POSITION

The Forward model was significant and had an adj rsquare of .222. However, all the variables were significant and produced an equation of BestTWPos = 51.1067 + .5457*(EntryPos) - 4.3556*(DurationDecimal) - 41.7290*(RepeatedRatio). Output 7 shows these figures.

			Analys:	is of Va	riance				
Source		DF		Sum of quares	Mean	Square	F	Value	Pr >
Model		3		104452		34817		50.57	<.000
Error		517	:	355923	688.	438128			
Corrected	Total	520		460375					
	Model	l Fit St	atistics						
R-Square	0.2	269	Adj R-Sq		0.2224				
AIC	3408.4	1202	BIC 341		10.4939				
SBC	3425.4	1432	C(p)		3.2332				
	Ar	nalysis	of Maximu	n Likeli	hood Est	imates			
					Standard	L			
Parameter		D	F Estin	nate	Error	tV	alue	Pr	> t
Intercept			1 51.	1067	11.9187		4.29	<	.0001
Intercept EntryPos				1067 5457	11.9187 0.0480		4.29 1.38		.0001 .0001
-				5457		1		<	

Output 7. Forward with Entry Position model output.

ASSESSING THE PIVOT MODELS

The main method for assessing the models was Adjusted R-square for continuous targets and misclassification rate for binary variable. Because many of the models performed poorly we choose the decision tree for classification and the Forward with entry date measuring BestTWPos. The decision tree had the best misclassification rate of 37%. The BestTWPos had .222 Adjusted R-square. The strengths of the decision tree model used was that it is easy to interpret and our model has a good performance in determining which songs will NOT reach the top 40.

While a common weakness of decision trees is that they can over-fit models, we were careful to omit variables with high correlation/collinearity prior to running the model. We also subjectively determined there were not too many "branches" of the tree, indicating it did not result in an over-fitting model. The biggest weakness of our resulting model was that the model has poor performance in identifying which songs will reach the top 40.

The Linear regression model has a very low adjusted r-square value. While the model is significant, its performance is poor and different variables (which are not available to us, such as marketing costs, etc.) would be needed to improve this model.

CONCLUSION

Our best model explained very little variance, indicating a need for additional variables that may be better predictors of song success on the Billboard Top 100 chart. Although our models explained little variance, knowledge of the trends over time are useful tools along with the decision tree analysis to allow for quick classification. Additionally, we may be unaware of any underlying biases or assumptions about our variables that led us to model them in the way we did. It is possible that our existing dataset could be analyzed to determine a complexity-popularity relationship within the songs on the Billboard Top 100, using a different combination of variables or modeling methods. However, our models, as they are, may still have additional benefactor value.

Music industry analysts may be able to benefit from our analysis by incorporating our data into their models, by inclusion of sales data and marketing investments in their catalogue of popular music. Additionally, there is strong evidence that private popular music enthusiasts and researchers may wish to utilize the data we collected to further their research and hobbies (as indicated by the existence of the Whitburn Project).

Our final recommendation is to seek additional data and variables that may impact results, incorporate them into the existing models and/or create new models to develop a predictive model that better explains popularity and lyric complexity.

Furthermore, the results of our analysis indicate the position at which a song enters the charts is a much better predictor than the complexity of lyrics in determining where the song will peak on the charts and how long the song will remain on the charts.

REFERENCES

Heaton, Michelle and Paris, Kelly. 2006. "The Effects of Music Congruency and Lyrics on Advertisement Recall." *UW-L Journal of Undergraduate Research*, Volume IX.

Powell-Morse, Andrew. "Lyric Intelligence In Popular Music: A Ten Year Analysis." *Seat Smart*. May 18, 2015. Available at http://seatsmart.com/blog/lyric-intelligence/.

Chang, Bettina. "Can a Song's Lyrics Predict Its Commercial Success?" *Pacific Standard*. March 19, 2014. Available at http://www.psmag.com/business-economics/can-songs-lyrics-predict-commercial-success-76936.

Barnes, Tom. "Pop Music Is More About Advertising Now Than Before — And Nobody Realizes It." *Music.Mic.* May 22, 2015. Available at http://mic.com/articles/118974/pop-music-is-more-about-advertising-now-than-before-and-nobody-realizes-it#.jUQOOR1eD.

ACKNOWLEDGMENTS

Dr. Goutam Chakrabory, Oklahoma State University

We would like to thank "Dr. C" for the important role he played in this project. Throughout the past two years he has provided us with the knowledge and resources to allow us to successfully undertake and complete this project. His guidance and teachings demonstrate the importance and proper steps in identifying a research opportunity and conducting the appropriate method of data collection and analysis. Dr. C's encouragement of continuous learning motivated us to develop new skills and created a hunger to conduct research, to explore new programs, to attend conferences, and to further test our abilities as we explore the world of analytics.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

John M. Harden Oklahoma State University John.Harden@Okstate.edu

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.