Music Genre Classification with Lyrical Features

SYS 6018

Andrew Evans (ace8p)

Dec. 11, 2018

Problem Background

Although music is frequently played in public and private venues, listeners are not always made aware of the song title or artist associated with the music they hear. Numerous websites and applications attempt to provide ways to identify a song for a user, but there remains opportunity to optimize song search using lyrics because the inherent nature of music to have high repetition and similar emotional sentiments from song-to-song within the lyrics makes identifying individual songs challenging. One way to improve search is to use a heuristic that identifies the most relevant results[5]. For song identification using lyrics, narrowing search results by genre could improve the speed and accuracy of a search query. The problem that must be solved is building a model using a data mining techniques which can produce an accurate classification of genre which could be used by organizations that offer song search to end users.

To properly simulate the features known about a song when searching, the only data that will be used for this data mining project is the lyrics of the songs. The lyrics data that will be used is a set of 380,000 songs with the lyrics scraped from MetroLyrics.com. The data includes the 'genre' label, which is missing from other singular sources of lyrics. A source of error for this data set could come from mislabeling of songs' genre. Each artist on this website is associated with one genre. It is possible for artists to release songs or entire albums that are classified as a genre separate from their past work, but the data represents no such scenario. Any songs of a different genre than what they are labeled as will train a model with error in the lyrical features and be harder to classify as part of a 'test' data set.

Project Objectives

This project seeks to optimize the accurate classification of music genre on a per-song basis. This can be measured by a true/false classification percent, a confusion matrix, or an ROC curve. For accurate classification, the percent of misclassifications would be low and the area under the ROC curve would be large. The accuracy of classification for both the 'train' and the 'test' set will be evaluated.

The project will also simulate the use of the model as a search tool heuristic by creating a function that takes a string of lyrics and a genre classification and return the predicted genre and an indication that the prediction was true or false.

Related Work

This project is not the first attempt to classify songs using lyrical features. Several other approaches have been taken using varying data sets, classification levels, and methods.

Ram (2017)[1] used a 55,000 song data set from LyricsFreak.com. Naive Bayes, Support Vector Machines (SVM) and a Neural Network were developed to classify songs into (3) genres. The features used were 539 words selected by FFT classification. The best train and test accuracy (94.4%, 93.2% respectively) resulted from SVM.

Hedayati (2018)[2] used 100 individual songs from SongLyrics.com. The approaches used were Decision Trees and Random Forests to classify songs into (6) genres. The Decision Tree approach proved the most accurate with 63.4% accuracy.

Liang et. al (2011)[3] used what is dubbed the "Million Song Dataset" with (10) possible genres for classification. There was a blended model used, with audio and lyrical features. The lyrics-only model (most similar to this project) used bag-of-words selected features and was trained on a multi-class logistic regression model which produced 31.4% accuracy.

The problem remains unsolved because of the additional potential for accurate classification across greater numbers of genres. It is a challenge because of the lack of lexical diversity in song lyric documents, resulting in term frequencies and inverse document frequencies that are not highly discernible from one document to the next.

Approach

This project operates on the hypothesis that songs of genres that are niche will be more accurately classified than popular genres. The popular and niche genres are defined by the amount of song consumption in 2017[4].

There are many hypotheses to evaluate due to the number of genres present in this data. Examples of these hypotheses are as follows:

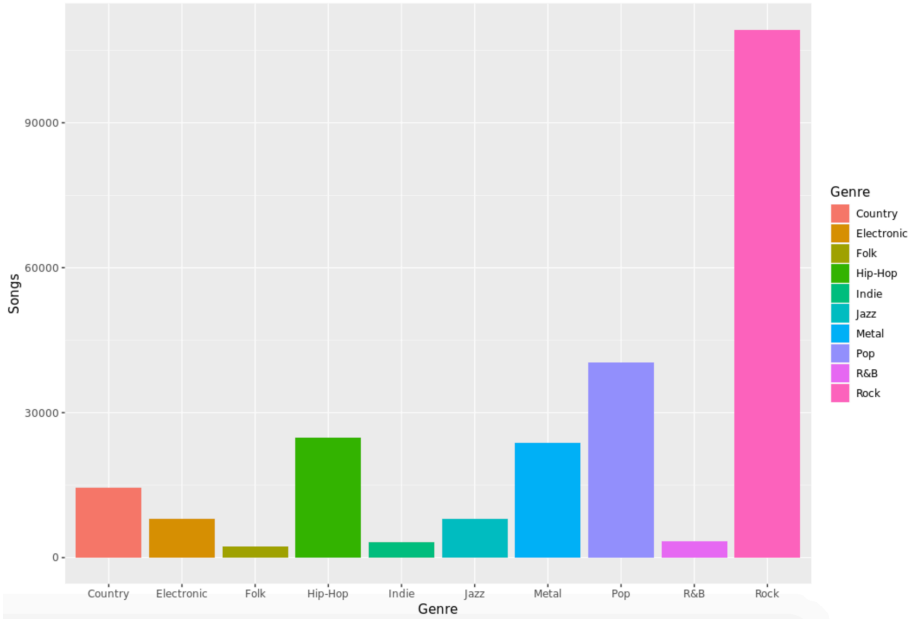
- The Metal genre will have a more optimal ROC curve than Pop.
- The Jazz genre will have a lower misclassification percent than Hip-Hop.

Like some of the related work above, this project will use multiple data mining methods including text mining for a TF-IDF matrix and sparse words removal as feature selection, followed by modeling with Random Forests and SVMs. Beyond those approaches, a lasso regression will be performed for additional feature selection, with a new set of features to use to train the RF and SVM models.

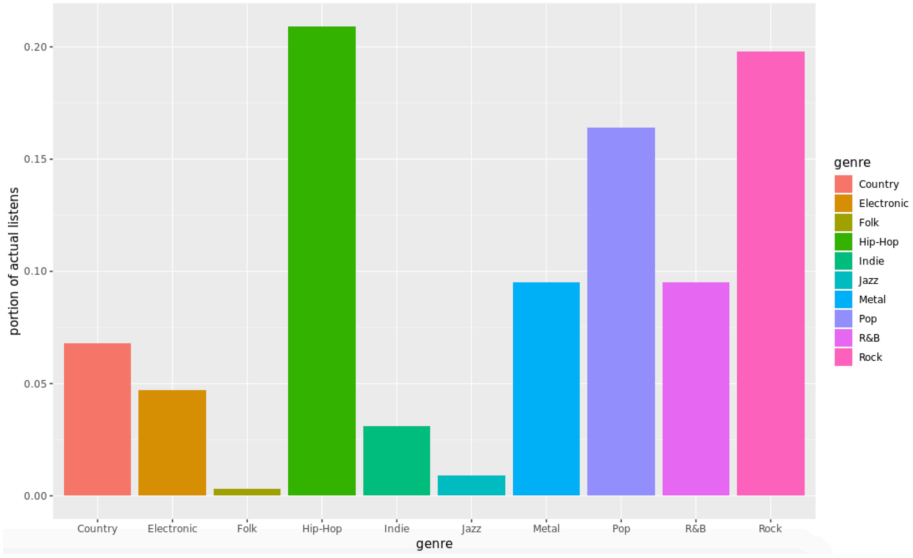
This methodology is not distinctly novel compared to related work and other data mining projects. It is instead an application of techniques to new data, with more challenging objective (read: more genres for classification) than what has already been done.

Execution and Results

Initial data exploration was performed to get a better understanding of the data prior to modeling. A look at the number of songs by genre in the data set revealed a much larger presence of Rock songs than other genres.

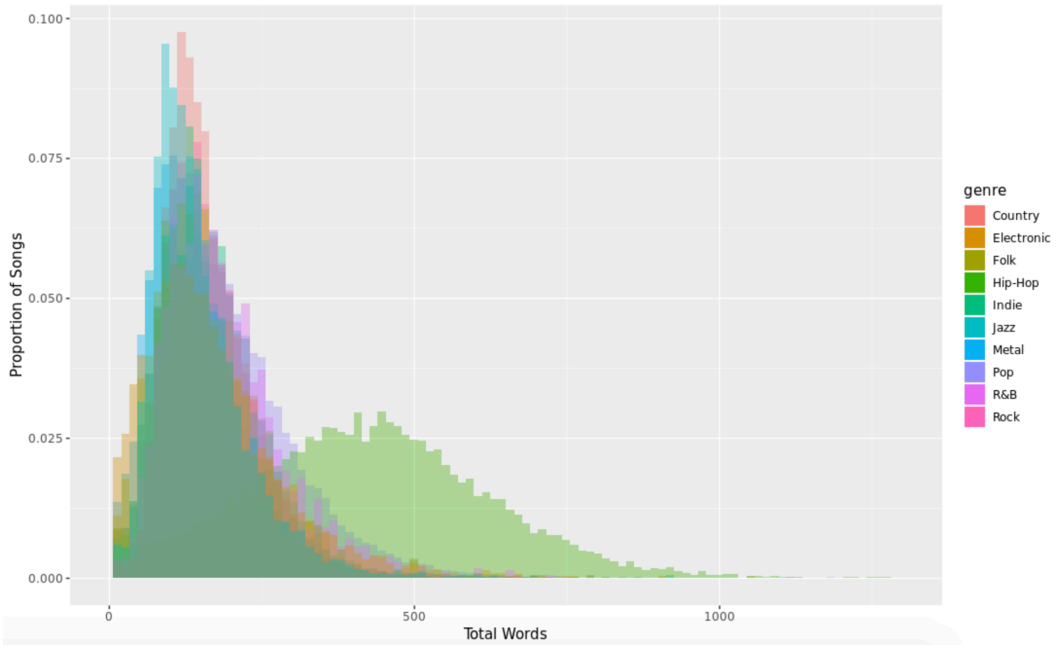


This differed from data on actual song ubiquity in listens by the music-enjoying populous, which indicated a greater representation of other genres, particularly Hip-Hop and Pop:

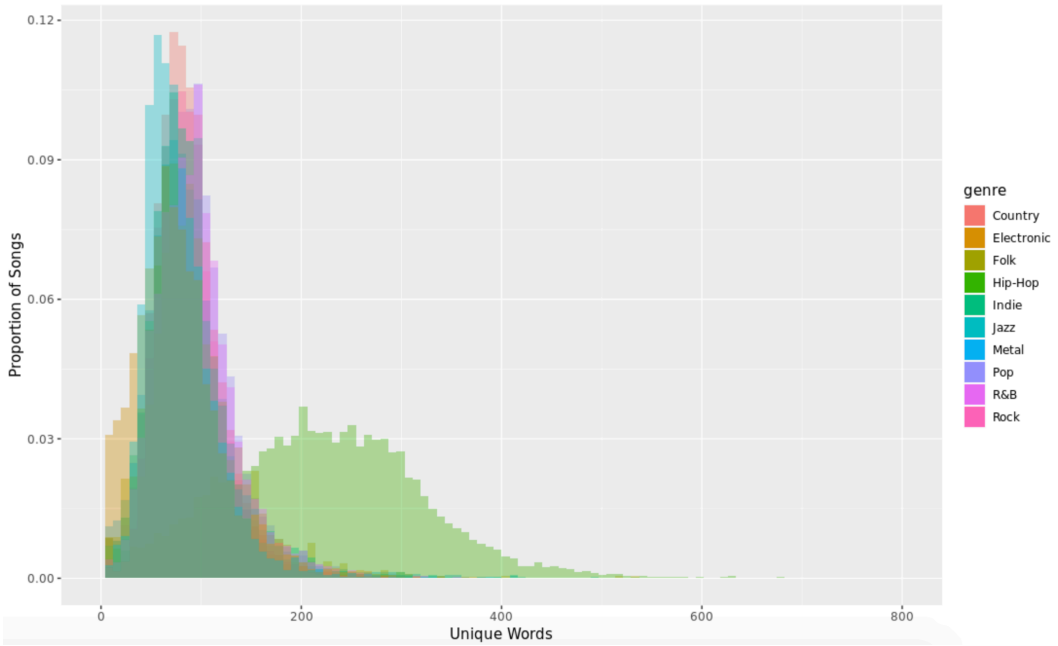


The data set's imbalanced classification presence could introduce error of False Positives or False Negatives related to the recognition of Rock only due to its ubiquity in the data set.

The features of the data set (the lyrics) were also explored. Histograms of the total words per song were generated by genre. It was clear that Hip-Hop songs were more verbose, and Pop songs had a slightly wider distribution of words than the rest of the genres which were largely similar.



The unique words by song were also displayed, which revealed a significant portion of Electronic songs with very few terms, while Hip-Hop still stood out with a significant amount more unique words per song.



The data exploration indicated that the hypothesis that more niche genres than Hip-Hop could prove false because of the unique nature of that genre's use of lyrics in both unique terms and term frequencies.

Prior to viewing the data by genre, rows of the data set with no lyrics and rows categorized as "Other" or "Not Available" were removed. These rows did not represent a use-case that this problem seeks to solve, and thus were not relevant to the model.

Text mining was performed on the lyrics data. After a corpus was developed using all documents in the data set and the corpus was modified to exclude punctuation, numbers, capital letters, and stop words, a document term matrix was formed with a TF-IDF weighting. A comparison was made between the removal of sparse terms at a 0.98 and a 0.95 threshold. The former threshold resulted in 571 terms while the latter resulted in 242 terms. A significant number of words in the broader set of columns were hypothesized to be important because they may not be heard in some genres, and thus the final feature set was a TF-IDF matrix of 571 terms. The list of terms separating these two thresholds is available in Appendix A.

Three modeling methods were chosen: Random Forest, Lasso Regression, and SVM. Model selection was based on the goal of avoiding overfitting to the data set with an imbalance of classes (predominantly Rock). Random Forest was selected because of its ability to handle a large 'n' and 'p' without *necessarily* overfitting by using a high number of trees. Lasso Regression was selected because of its nature to reduce coefficients to 0 for improved performance in prediction. The size of coefficients was also important for attempting to select a subset of important features to improve the performance of the other models. SVM was chosen because it performed well in the related work and is known for strong performance in classification due to the 'kernal trick' wherein the features are mapped to high-dimensional feature spaces.

The Lasso Regression produced a set of coefficients to examine for more understanding of their impact on classification to specific genres. Total magnitude and standard deviation of the coefficients across genres was calculated and the highest of each were viewed.

	Rock	Country	Electronic	Folk	Hip.Hop	Indie	Jazz	Metal	Pop	R.B	Tot	Sd
nigga	-30.957528874	-14.975060	-3.9529956	0.0000000	27.61664962	0.0000000	0.000000	14.688749	5.3376048	12.0843677	109.61296	16.129758
rap	-13.210666824	-15.485820	10.9978008	-7.8980462	21.61448036	3.4709717	-14.029554	0.000000	2.1908504	0.0000000	88.89819	11.847515
gon	-18.893099080	-22.773052	2.2850739	-0.7020430	6.97829794	0.0000000	4.141508	-8.807598	3.4664139	3.4127290	71.45982	10.289858
fuck	2.460258453	-21.564639	2.2145364	-7.1772012	4.06502338	0.0000000	-8.667313	5.187084	-4.4804923	4.5156065	60.33215	8.376847
nothin	0.000000000	2.273957	-3.5431383	0.0000000	0.07912302	-3.8987359	1.377914	-9.521819	-2.1060732	0.4546582	23.25542	3.463835
town	0.944580795	3.468702	-1.3832397	2.8508406	-4.12467807	0.8616495	3.207887	-4.845011	-1.5827912	0.0000000	23.26938	2.924111
bar	2.233830407	6.650129	-2.2754084	0.0000000	2.71059305	-0.9739127	1.638315	-1.595799	-2.8288540	-2.4079415	23.31478	2.994920
style	-2.105462045	-1.703882	0.2982715	5.7222537	5.47269930	-4.4273632	1.673283	-2.350569	0.0000000	0.0000000	23.75378	3.289396
crack	1.076900533	-2.365121	-0.6647212	3.2769181	3.51046698	1.4861983	-2.949737	2.622932	-2.3988883	-3.5949478	23.94683	2.722841
dead	1.242052345	-4.374702	-2.9856152	2.3609374	0.25767217	0.0000000	-1.733305	4.183229	-5.0553259	2.4150418	24.60788	3.086996
babi	0.000000000	-1.268262	0.0000000	-2.7543599	1.85780837	-6.6224773	1.511777	-5.452821	1.8583663	4.4196891	25.74556	3.453645
love	-1.653624953	2.905647	0.5391584	-0.9738287	-1.48187884	0.0000000	3.158964	-11.554257	1.5545311	2.3484015	26.17029	4.260469
brain	2.289130234	-2.619059	1.1693122	-6.1741327	2.31761261	0.4952431	0.000000	3.861940	-5.4656521	-1.9891972	26.38128	3.369951
wit	0.000000000	-4.021074	0.0000000	-0.2993771	6.87159717	-5.1822882	1.322109	4.365260	2.4519648	-2.2220910	26.73576	3.666774
aint	-0.400905853	3.847877	-4.0116830	0.0000000	4.48085238	-1.6316555	2.879699	-7.704353	0.0000000	2.9216777	27.87870	3.800178
caus	-1.765174339	1.830276	-0.8881161	-3.6771560	7.14304245	0.9103543	-2.051682	-5.585565	3.0490711	1.0349505	27.93539	3.627115
just	-0.296229414	4.753308	-1.2699417	-8.6404646	1.24275442	0.0000000	2.481070	-4.816793	0.1497502	4.6804034	28.33072	4.102679
yeah	2.662148622	-1.518809	2.2232852	-4.3439019	2.27729006	-1.0855490	-1.959431	-5.646382	3.0867059	4.3046439	29.10815	3.388698
ass	1.691594600	-3.320037	2.0010954	-0.5723578	7.19343573	0.0000000	-7.824398	7.380195	0.0000000	-0.3507031	30.33382	4.496054
tryin	0.005120114	2.628146	-4.0450336	0.0000000	4.32343319	-10.0131409	1.871648	-3.803564	-0.5846205	3.3251913	30.59990	4.315778
old	-0.087987303	5.693006	-4.5259157	2.5503856	-5.62376723	2.4329690	3.403699	-2.046569	-3.1914730	1.3956525	30.95142	3.724685
well	1.349743518	2.634885	-4.6103586	1.2131888	-10.19813278	1.6285546	0.000000	-5.080041	-4.3387396	0.0000000	31.05364	4.116337
chorus	-0.073884648	3.925841	-3.8097953	2.3078081	2.99619145	-10.1195805	-2.721677	0.000000	2.8449470	3.2792743	32.07900	4.380617
scream	1.593292705	-9.711120	0.2936670	-2.5129901	0.31102627	0.0000000	-12.787707	3.938512	-1.0889786	0.2021764	32.43947	5.212724
lookin	1.963766499	4.006107	-11.1264092	-1.6248779	4.04278786	-2.2854191	0.000000	-7.467197	1.5322109	1.5333017	35.58208	4.938320
bleed	3.659735767	-7.466837	-2.0918531	0.0000000	-1.92780138	1.9104928	-10.765150	5.809547	0.0000000	2.3225248	35.95394	5.039994

Several of the terms with high magnitude coefficients were kept when choosing the 0.98 threshold versus the 0.95 threshold, implying that the selection could have been the better choice.

The Random Forest and SVM models were trained with both the 'full' feature set of 571 words as well as the subset of features that had a higher total magnitude of coefficients from the Lasso Regression.

Full Model:

True Classification %	Train	Test
randomForest	60.6%	60.9%
cv.glmnet	55.3%	54.9%
svm	51.3%	50.5%

Lasso-Selected Reduced Model:

True Classification %	Train	Test
randomForest	60.3%	60.8%
svm	52.1%	50.6%

The Random Forest model performed best for Train and Test using both the Full Model and the Reduced Model. The Test fit was slightly better than the Train fit. The Reduced Model feature set did not improve the Random Forest fitted values or the predicted values, although the Test fit continued to outperform the Test fit, by a larger margin. The Lasso-selected features did improve the SVM model, but not by an amount that affected the choice of model.

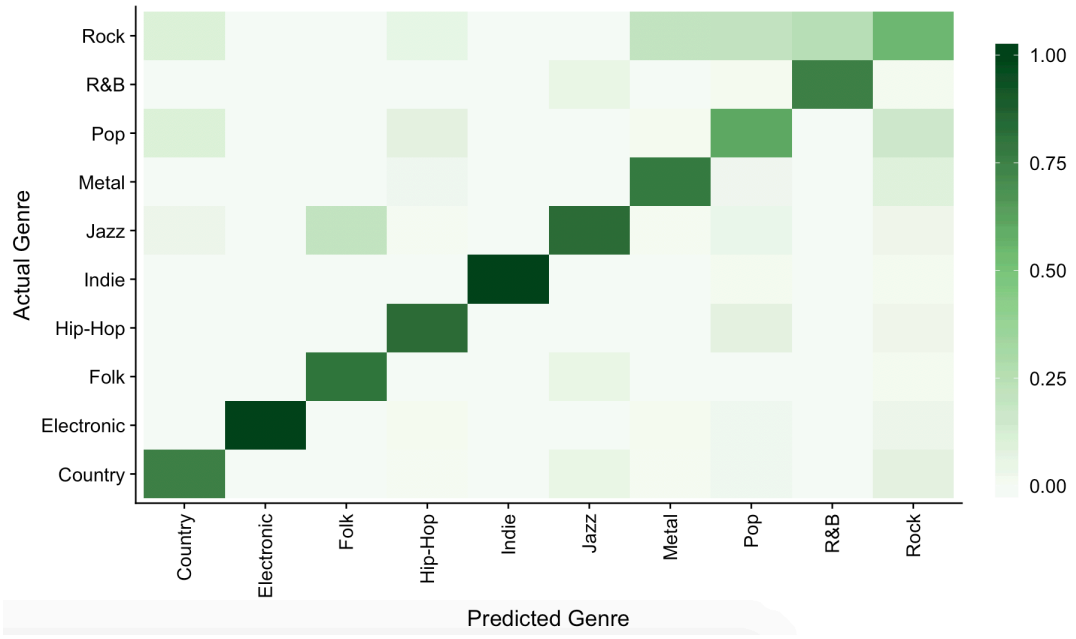
One last model was developed using the best method (Random Forest) on a subset of rows of the data, for the (3) most prominent genres: Rock, Pop, and Hip-Hop. This was developed as a comparison to the related work in which Ram (2017) achieved 93.2% accuracy.

3-Genre Model with Lasso-Selected Features:

True Classification %	Train	Test
randomForest	85.3%	82.8%

The model developed for this data when only (3) genres were present did not perform as well as the SVM model that Ram (2017) used for a separate data set. Additional modeling would need to be done to evaluate if that related work's accuracy could be achieved with this new data.

To test the hypothesis, further evaluation was performed on the classification by genre. The following represents the strongest performing model (RF, Full) on the Test dataset

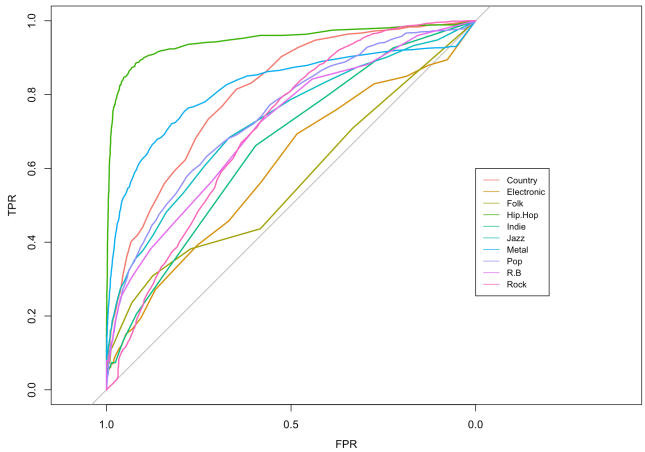


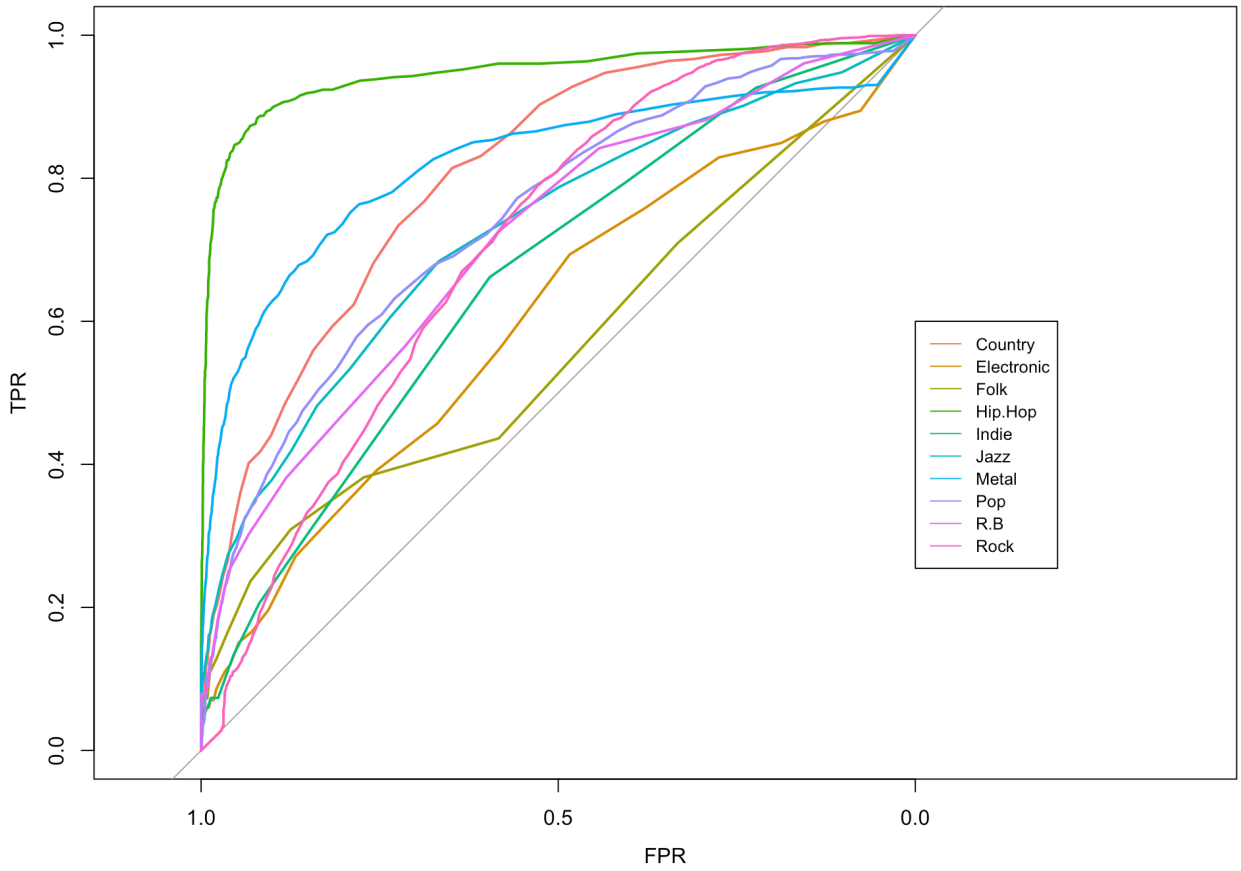
A confusion matrix did indicate strong performance for niche genres like Electronic and Indie, but already provided evidence against the hypothesis in that Hip-Hop performed distinctly better than Rock and Pop.

For the hypothesis to be true in general, the performance by genre would go in order:

Best Worst
Folk > Jazz > Indie > Electronic > Country > R&B > Metal > Pop > Rock > Hip-Hop

Although Rock and Pop did underperform most genres, Hip-Hop and Metal outperformed the expectation. The final step for evaluation of the hypothesis was to view the ROC curve for each genre. (*see enlarged ROC curves on next page*)





Although the stated hypothesis, “Metal will have a better ROC curve than Pop” is true (as it has more area under the curve), it again appears that in not all cases the less popular genre performed better than the more popular genre.

I reject the hypothesis, now understanding that the popularity of the genre can sometimes indicate its predictability, but not always.

Conclusion

The problem of classifying songs based on lyrics alone remains unsolved. Text data for music proves to be a challenging feature set to classify based upon due to the high-dimensional nature and similarity between documents. Particularly in this scenario, where the data set had an imbalance of classes, modeling was highly sensitive to the over-classification of the predominant genre, Rock, which saw the worst prediction performance. My experience indicates that the inclusion of audio features, as used by Liang et. al[3], or other features beyond the lexical characteristics of a song are key to achieving fast, accurate identification of songs and their genre.

Appendix A

Terms separating the 98% and 95% threshold of sparse terms removal.

"across"	"act"	"afraid"	"air"	"aliv"	"along"	"alreadi"	"alright"	"angel"	"answer"	"anymor"	"anyth"
"apart"	"arm"	"ass"	"ball"	"bar"	"beauti"	"becom"	"bed"	"begin"	"bit"	"bitch"	"blame"
"bleed"	"blind"	"block"	"blow"	"bone"	"born"	"bout"	"brain"	"bridg"	"bright"	"broke"	"broken"
"brother"	"build"	"buy"	"car"	"carri"	"catch"	"caught"	"chain"	"chanc"	"check"	"child"	"children"
"citi"	"clear"	"cloth"	"cloud"	"comin"	"control"	"cool"	"corner"	"couldnt"	"count"	"cover"	"crack"
"crazi"	"cross"	"crowd"	"cut"	"daddi"	"damn"	"deal"	"dear"	"death"	"differ"	"doesnt"	"dog"
"doubt"	"dress"	"drink"	"drive"	"drop"	"ear"	"earth"	"easi"	"eat"	"els"	"empti"	"everybodi"
"everyon"	"fade"	"faith"	"famili"	"fast"	"father"	"feet"	"fell"	"felt"	"fill"	"final"	"fine"
"finger"	"flame"	"floor"	"flow"	"follow"	"fool"	"four"	"front"	"fuckin"	"full"	"fun"	"futur"
"gave"	"gettin"	"goe"	"goin"	"gold"	"gon"	"goodby"	"great"	"green"	"ground"	"grow"	"guess"
"gun"	"guy"	"hair"	"half"	"hang"	"happen"	"happi"	"heat"	"heaven"	"honey"	"hot"	"hour"
"hous"	"human"	"isnt"	"jump"	"key"	"kick"	"kid"	"kind"	"king"	"knee"	"knock"	"known"
"ladi"	"land"	"late"	"laugh"	"lay"	"lead"	"learn"	"lip"	"lock"	"lone"	"lookin"	"lord"
"lot"	"lover"	"low"	"mad"	"mama"	"matter"	"may"	"meant"	"meet"	"memori"	"men"	"met"
"mile"	"million"	"minut"	"moment"	"moon"	"morn"	"mother"	"mouth"	"music"	"near"	"next"	"nice"
"nigga"	"nobodi"	"nothin"	"number"	"ooh"	"outsid"	"part"	"parti"	"past"	"pay"	"peac"	"perfect"
"phone"	"pick"	"pictur"	"piec"	"plan"	"point"	"pop"	"power"	"pray"	"pretend"	"pretti"	"pride"
"promis"	"pull"	"push"	"que"	"question"	"quick"	"quit"	"rais"	"rap"	"reach"	"read"	"readi"
"realiz"	"reason"	"red"	"repeat"	"rest"	"ring"	"rise"	"river"	"road"	"rock"	"room"	"round"
"rule"	"sad"	"save"	"saw"	"scare"	"school"	"scream"	"sea"	"search"	"second"	"secret"	"send"
"sens"	"shadow"	"shake"	"share"	"shoe"	"shoot"	"shot"	"sick"	"sight"	"sign"	"silenc"	"sin"
"sinc"	"skin"	"slip"	"slow"	"smoke"	"somebodi"	"sometim"	"somewher"	"son"	"soon"	"sorri"	"space"
"speak"	"spend"	"stare"	"step"	"stick"	"stone"	"stori"	"straight"	"strong"	"style"	"summer"	"tast"
"thank"	"theyr"	"thousand"	"three"	"throw"	"tight"	"til"	"tire"	"today"	"tomorrow"	"top"	"track"
"train"	"tree"	"troubl"	"trust"	"truth"	"tryin"	"upon"	"vers"	"voic"	"wake"	"wall"	"war"
"warm"	"wasnt"	"wast"	"water"	"wave"	"weak"	"wear"	"went"	"weve"	"whatev"	"whisper"	"white"
"whos"	"wild"	"win"	"wind"	"window"	"wing"	"wit"	"within"	"woman"	"worri"	"worth"	"wouldnt"
"write"	"yall"	"yet"	"youd"	"young"							

References

[1] “Using Song Lyrics and Frequency to Predict Genre.” [Online]. Available: http://cs229.stanford.edu/proj2017/final-reports/5241796.pdf.

[2] Hedayati, R. (2018). Can a song's lyrics predict it's genre? (part 1). [online] Riazhedayati.github.io. Available at: https://riazhedayati.github.io/blog/predict-song-genre-pt1/ [Accessed 2018].

[3] D. Liang, H. Gu, and B. O'Connor, “Music Genre Classfiication,” Columbia. [online]. Available at: http://www.ee.columbia.edu/~dliang/files/FINAL.pdf.

[4] “Hip Hop Has Replaced Rock As Music's Most Consumed Genre,” Digital Music News, 05-Jan-2018. [Online]. Available: https://www.digitalmusicnews.com/2018/01/04/hip-hop-rock-2017-biggest-genre/.

[5] M. Kessentini, H. Wang, J. T. Dea, and A. Ouni, “Improving Web Services Design Quality Using Heuristic Search and Machine Learning,” 2017 IEEE International Conference on Web Services (ICWS), 2017.