What's the Next Hip Hop Hit?

December 7, 2018

Bofei Zhang (bz1030) Shaoling Chen(sc6995) Ziyu Lei(zl2350) Zian Chen(zc674)

Contents

1 Business Value

What makes a certain hip-hop song a hit? Is there any similarity shared by these Billboard winners over time? The aim of the project is to identify the elements that make up a popular hip-hop song. More specifically, by looking at the lyrics, performing sentiment analysis, and other analytical modelling, we would like to know if a given song would become a hit on the Billboard charts. The practical value of the project lies in the fact that entertainment industry is a typical "winners-take-all" market. Technology may seem to enable people to have access to a wider range of music options but it actually skews people's consumption of the biggest hits and most powerful platforms. It turns out everyone wants hits — the more familiar the better, says Derek Thompson, author of a book entitled 'Hit Makers'.[1] Determining the formula for hit songs will give musicians of this generation insights into composing and producing songs that have high business value.

In this project, we used several models including Bernoulli Naive Bayes, Logistic Regression, and Random Forest. Sentiment analysis and topic modelling were as well applied. We also explored the mechanism behind the models we tested in order to explain the accuracy level observed.

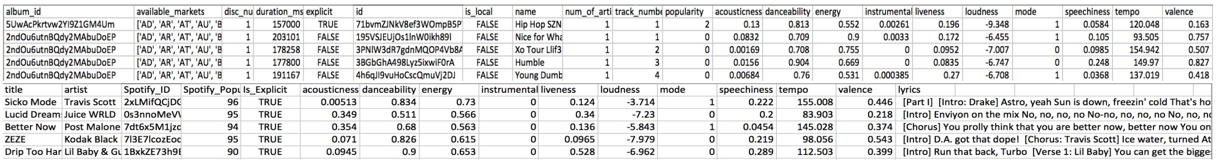
2 Data Extraction and Pre-processing

2.1 Data Collection and Target Variable

Data was sourced from three major platforms, Spotify, Billboard and Genius. In oder to get the most basic information from Spotify, setting up Spotify user account and registering application for data request from Spotify Web API was first step. We then generated access token by the client id and client secret key. Album information was retrieved by searching the key word "Hip Hop". This, however, only gave us the albums that matched the key word. We then called the Spotify API again to search the tracks in the albums from the prior step. Each album and song was assigned a unique ID by Spotify. We saved the dataframe with track information as `tracks.csv`.

More information was available through the Spotify API. We also included the audio features of each track in `tracks.csv`. We called the Spotify API by the tracks' unique ID, received audio feature object in JSON format and scraped all the feature variables to add to the orginal csv file. Added features included popularity, acousticness, danceability, energy, instrumentalness, liveness, loudness, mode, speechiness, tempo and valence. The specific explanation of all these features can be seen here. The new dataframe was saved as `tracks_with_audio_features.csv`.

Billboard Hot R&B/Hip Hop weekly chart (with 50 songs on it) is the source page that we crawled. We considered Billboard ranking a reliable measurement of a song's popularity because it combined airplay, social media and streaming as well as album



album_id	available_markets	disc_nu	duration_ms	explicit	id	is_local	name	num_of_arti	track_numbe	popularity	acousticness	danceability	energy	instrumental	liveness	loudness	mode	speechiness	tempo	valence
5UwAcPkrtvw2YI9Z1GM4Um	['AD', 'AR', 'AT', 'AU', 'B	1	157000	TRUE	71bvmZJNkV8ef3WOmpB5P	FALSE	Hip Hop SZN	1	1	2	0.13	0.813	0.552	0.00261	0.196	-9.348	1	0.0584	120.048	0.163
2ndOu6utnBQdy2MAbuDoEP	['AD', 'AR', 'AT', 'AU', 'B	1	203101	FALSE	195VSJEUjOs1lnW0ikh89l	FALSE	Nice for Wha	1	1	0	0.0832	0.709	0.9	0.0033	0.172	-6.455	1	0.105	93.505	0.757
2ndOu6utnBQdy2MAbuDoEP	['AD', 'AR', 'AT', 'AU', 'B	1	178258	FALSE	3PNlW3dR7gdnMQOP4Vb8A	FALSE	Xo Tour Llif3	1	2	0	0.00169	0.708	0.755	0	0.0952	-7.007	0	0.0985	154.942	0.507
2ndOu6utnBQdy2MAbuDoEP	['AD', 'AR', 'AT', 'AU', 'B	1	177800	FALSE	3BGbGhA498Lyz5ixwiF0rA	FALSE	Humble	1	3	0	0.0156	0.904	0.669	0	0.0835	-6.747	0	0.248	149.97	0.827
2ndOu6utnBQdy2MAbuDoEP	['AD', 'AR', 'AT', 'AU', 'B	1	191167	FALSE	4h6qJl9vuHoCscQmuVj2DJ	FALSE	Young Dumb	1	4	0	0.00684	0.76	0.531	0.000385	0.27	-6.708	1	0.0368	137.019	0.418

title	artist	Spotify_ID	Spotify_Popu	Is_Explicit	acousticness	danceability	energy	instrumental	liveness	loudness	mode	speechiness	tempo	valence	lyrics
Sicko Mode	Travis Scott	2xLMifQCjDG	96	TRUE	0.00513	0.834	0.73	0	0.124	-3.714	1	0.222	155.008	0.446	[Part I] [Intro: Drake] Astro, yeah Sun is down, freezin' cold That's ho
Lucid Dream	Juice WRLD	0s3nnoMeVV	95	TRUE	0.349	0.511	0.566	0	0.34	-7.23	0	0.2	83.903	0.218	[Intro] Enviyon on the mix No, no, no, no No-no, no, no, no No, no, n
Better Now	Post Malone	7dt6x5M1jzc	94	TRUE	0.354	0.68	0.563	0	0.136	-5.843	1	0.0454	145.028	0.374	[Chorus] You prolly think that you are better now, better now You on
ZEZE	Kodak Black	7l3E7lcozEoc	95	TRUE	0.071	0.826	0.615	0	0.0965	-7.979	0	0.219	98.056	0.543	[Intro] D.A. got that dope! [Chorus: Travis Scott] Ice water, turned At
Drip Too Har	Lil Baby & Gu	1BxkZE73h9E	90	TRUE	0.0945	0.9	0.653	0	0.528	-6.962	0	0.289	112.503	0.399	[Intro] Run that back, Turbo [Verse 1: Lil Baby] You can get the bigge

Figure 1: `tracks_with_audio_features.csv` and `billboard_with_lyrics.csv`

sales altogether when ranking the weekly chart. The reason we didn't use the popularity score feature is because the popularity scores on Spotify are extremely right skewed and it's hard to find a cutoff score to classify one song as popular or not. The target variable in this project, if a hip-hop song is a hit or not, is indicated by if it appeared on Billboard at least once or not. We scraped the info about the songs on the chart from year 2009 including the following metrics: Artsit (artist name), isNew (if this song is a new song), lastPos (the rank of last week), peakPos (the peak of rank), title(song title) and weeks(how long does it stay on billboard).We repeated similar process as before, retrieved the audio features of the songs on Billboard and saved the dataframe as `billboard_with_audio_features.csv`.

After getting the basic information of the hip hop songs we would like to study, We referred to Genius API to get the lyrics of the songs. We sent a GET request to the Genius API, iterated over the `hits` key in the returned object and tried to find the exact artist name match. After we found the exact match object, Genius returned a url and data needed to be pulled from the HTML file. In this case Beautiful Soup Library is a good tool to scarpe the lyrics. This process was repeated for songs from both `tracks_with_audio_features.csv`. and `billboard_with_audio_features.csv`. `billboard_with_lyrics.csv` that was saved after this process is included in Figure 1. Finally, we collected 2720 distinct songs from Billboard, and 69846 distinct tracks from Spotify. Since data was collected from multiple platform, lack of a unique key introduced extra difficulty to join the data set. We applied fuzzy string match to join dataset from Spotify, Genius, and Billboard with 0.5 similarity threshold computed by token sort ratio. To be accurate, we have dropped the track with missing or duplicate data. For example, searching lyrics by song name on Genius API may return multiple hits. We chose to drop this instance when we were unable to join it with data from other platform by fuzzy string matching. Full dataset includes 2201 tracks with positive label and 7593 tracks with negative label.

2.2 Data Cleaning

We classified the songs into two categories: positive and negative, grouped by the criteria if it appeared in the billboard weekly R&B/Hip Hop hits (at least once). Due to the nature of strong competition of Billboard, we only had small portion of track with positive label. Therefore, we under-sampled the negative class to generate a balanced dataset. After cleaning the data and random sampling, there were overall 2201 positive targets and 2201 negative targets in the dataset.

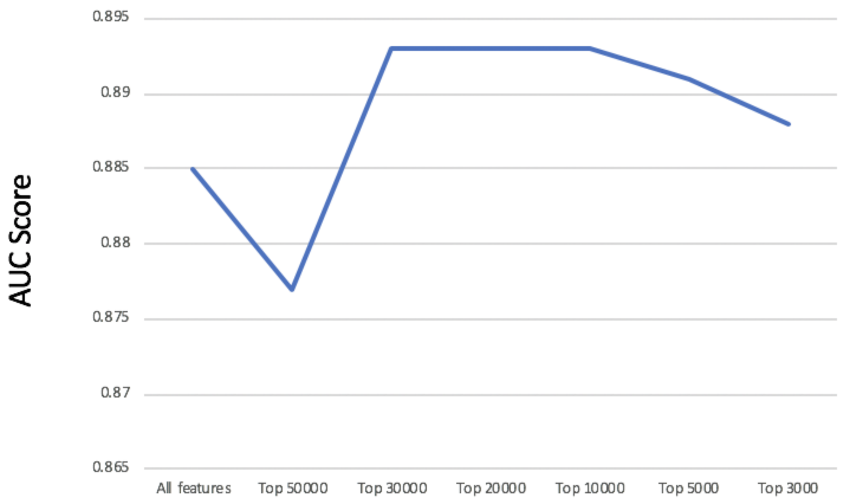


Figure 2: Baseline Model Performance Over Different Feature Sets

To tokenize lyrics better, we replaced the confusing punctuation like \n, digits and underscores. We also removed English stop words which do not have any meaning. Due to our data collection process, we found the lyrics crawled from Genius contained artists name within the brackets or parenthesis. Names of artist were strong predictor to our target variable, which led to higher performance for our model. Since lyrics is the primary topic of this project, we decided to drop all words inside brackets and parenthesis. Next, Count and TFIDF tokenizer was applied on the lyrics to generate training data. Parsing 4000 songs, 2-gram would generate over 2 million features, while 1-gram demonstrated a similar performance as 2-gram. Therefore,we only used 1-gram in the text mining process for simplicity. To further reduce dimensionality of this dataset, we applied stemming method Porter Stemmer to classify each word to their root. Since there are large portion of words that only appear couple times, we selected certain amount of most frequent words to train the model. Finally, as the Figure 2 indicates, top 10000 most frequent words in the training set is sufficient to generate same AUC score as the full dataset. But further reducing dimension of dataset would hurt the performance of model.

3 Data Visualization and Exploration

3.1 Sentiment Analysis

We started out to study the songs that appear on Billboards first by using the NLTK package in python. We tokenized the lyrics into words and calculated the frequency of the words through NLTK FreqDist module. Figure 3 and 4 show the words that occur most frequently among the hits. Then by implementing the built-in sentiment analysis module from NLTK, each sentence of the lyrics were assigned a compound score that indicates the overall connotation of its meaning (negative if compound<-

0.5, positive if compound>0.5, neutrality otherwise).The result shows that, there are about 4% more sentences with positive connotation than the ones with negative connotation; while the sentences with neutral meaning account for only about 5.3% percent of the overall set.

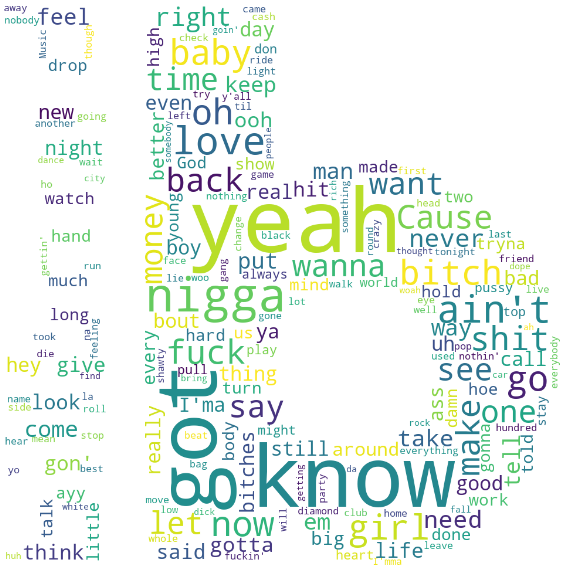


Figure 3: Hit Songs Lyrics Word Cloud

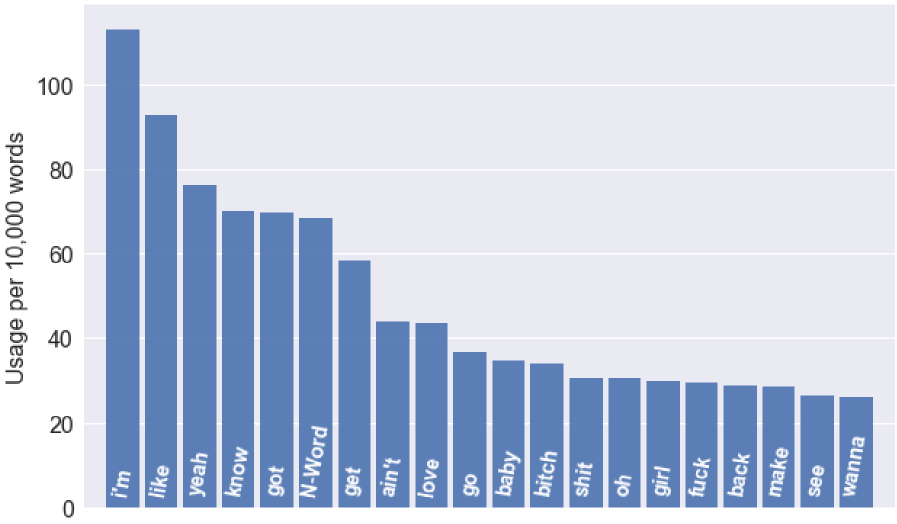


Figure 4: The plot of most frequently used words

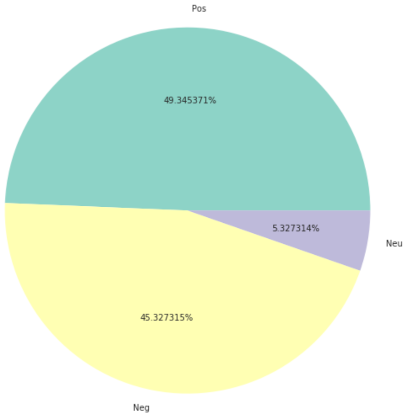


Figure 5: Overall sentiments

3.2 Topic Modelling

Topic modelling helps to extract information from large volumes of texts. It enables us to go further beyond sentiment analysis to gain a deeper understanding of what is specifically sung in hip-hop. We imported Gensim package and used the Latent Dirchlet Allocation (LDA) for this section. LDA considers each song's lyrics as a collection of topics in a certain proportion and each topic a collection of keywords in a certain proportion. [2] Again we limit our analysis to the songs that have appeared on Billboard.

We split the topics into 5 categories due to the fact that more topics will lead to topic bubbles that overlap while fewer topics will not be informative enough. Take a closer look at the three dominant topics (Figure 6). The first topic featured by the words "dont","know","like" and "love" can be related to the "love" theme apparently. The second topic featured by the words ("nigga","bitch","got","like" and "fuck") can be categorized as the vulgarity or misogyny in hop hop. The third topic that contains the collection of words like "girl"," babi" and "know" remind people of the joy in love, more concrete than the first topic. All these above inferences from the keywords are quite intuitive but give us a clearer picture of the regime/scheme of hip hop lyrics.

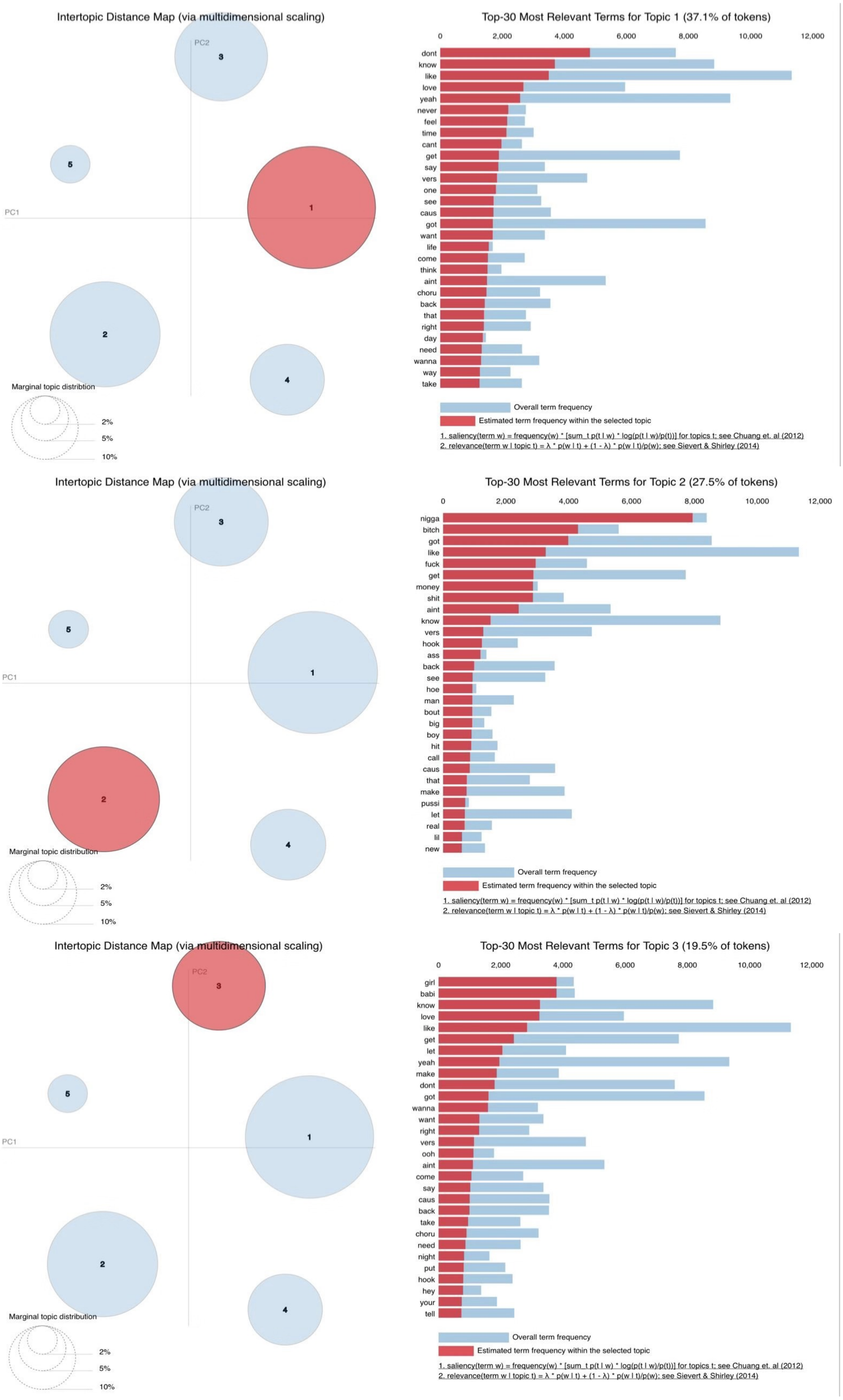


Figure 6: Topic Modelling

4 Modelling, Evaluation and Selection

In this section we provide a detailed analysis of all the models we applied to analyze the data.

4.1 Baseline Model and Evaluation

In the baseline model, we used Bernoulli Naive Bayes and Logistic Regression to fit the cleaned data sample due to the nature of the project: text mining. First we vectorized the lyrics by counter vectorization and tfidf vectorization (inverse document frequency measure) and transformed them into a matrix of binary indicators. Skilearn split the data into train and test set and the two models were applied to fit the data sample.

It turned out that Logistic Regression produced a better result. The Logistic Regression AUC score reached 0.883 for countvectorization and 0.900 for tfidf vectorization; while Bernoulli NB gave a super high recall but poor accuracy and precision. The reason Logistic Regrssion outperformed the Bernoulli Naive Bayes model is probably that Bernoulli NB makes the assumption that each feature is conditionally independent, which is very unlikely within the lyrics context. Over fitting becomes an issue in Bernoulli Naive Baye case. Research work in the machine learning provides support to the result we got. Ng and Jordan analyzed on discriminative and generative classifiers (namely, logistic regression and naive Bayes) and observed that generative naive Bayes initially does better, but discriminative logistic regression eventually catches up to and quite likely overtakes the performance of naive Bayes as the number of training examples m is increased.[3]

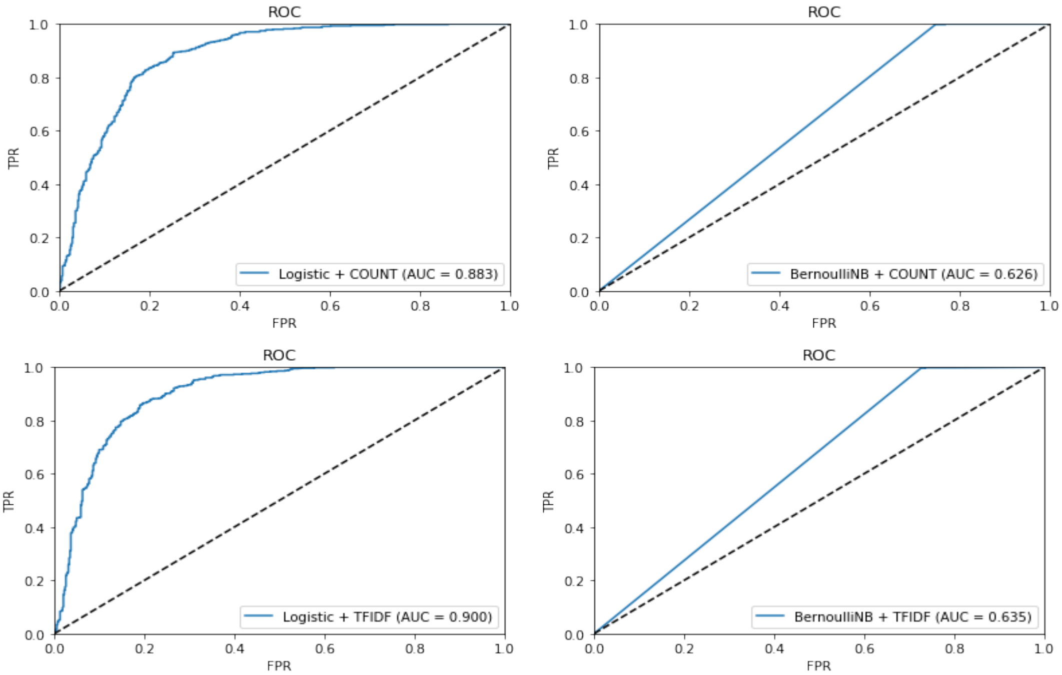


Figure 7: Baseline Model ROC

4.2 Cross Validation and Grid Search

We explored deeper into the Logistic Regression by carrying out grid search for the best parameter in the model. We imported GridSearchCV class from the $sklearn.modelselection$ library. We set the cross-validation folds as default number 3 and implemented the GridSearchCV. This step helped us find the parameter values that lead to the overall highest classification accuracy in the cross-validation: C (inverse of regularization strength)= 0.01 and $penalty = l2$ for the Logistic Regression with CountVectorizer model, $C = 10$ and $penalty = l2$ for the Logistic Regression with TFIDFVectorizer model. The AUC score increased from 0.883 to 0.923 and 0.900 to 0.951, respectively.

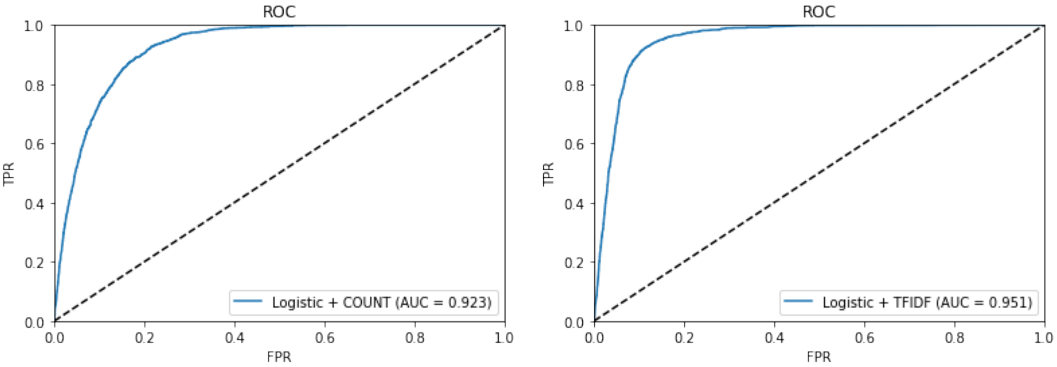


Figure 8: Baseline Model After Grid Search for Parameter Tuning

4.3 Model with Audio Features

Lyrics are one essential part of a song, but other features like rhythm, tempo and background instruments and so on are indispensable elements of it as well. Figure 9 shows the audio features' change of songs that appears on the billboard over years. So we planned to incorporate the intrinsic features that we retrieved from Spotify and build another classifier purely based on audio features. Random Forest model was applied in this model, as feature scaling (normalizing the data) was not required for this model. Random Forest Model based on audio features resulted in a pretty good AUC score 0.925. The top three audio features that contribute to the model result are: "instrumentalness","loudness" and "valence". The official descriptions of these three variables are respectively as follows:

Instrumentalness: Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal".
Loudness:The overall loudness of a track in decibels (dB).
Valence:A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

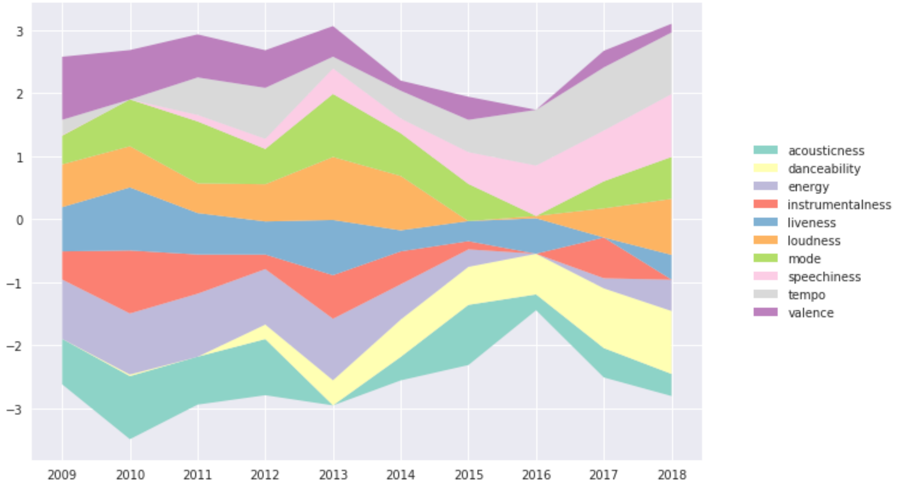


Figure 9: Billboard Songs Audio Features Changes Over Years

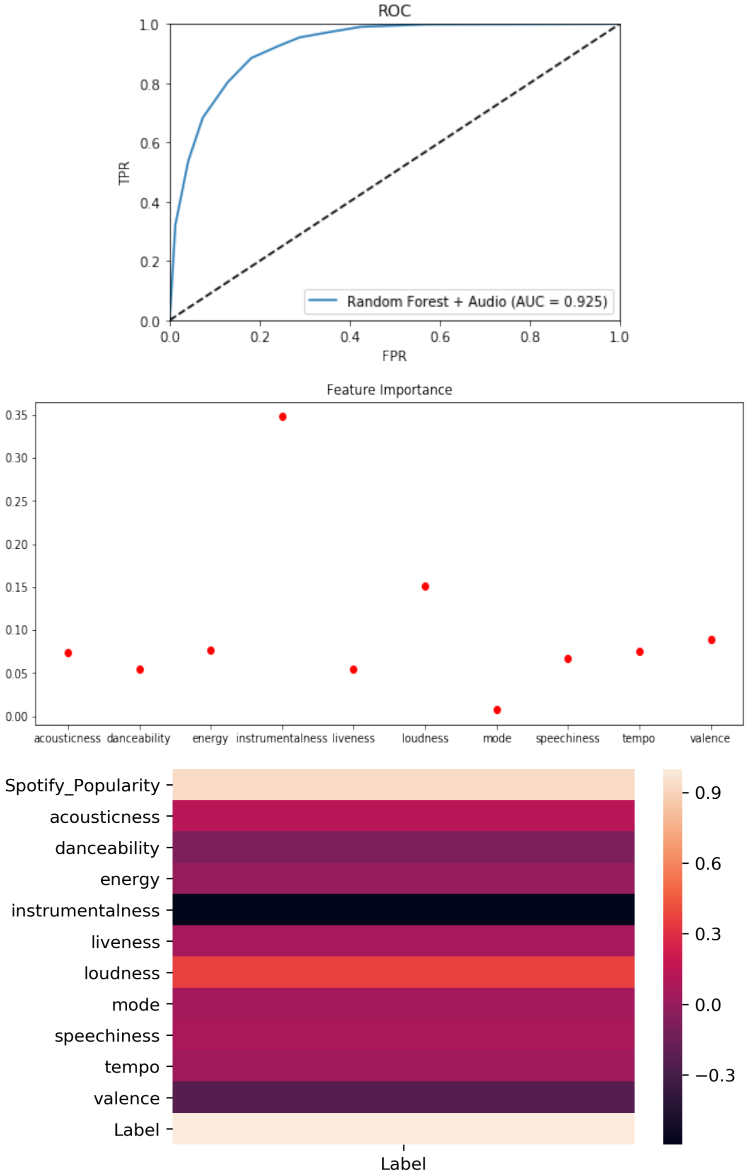


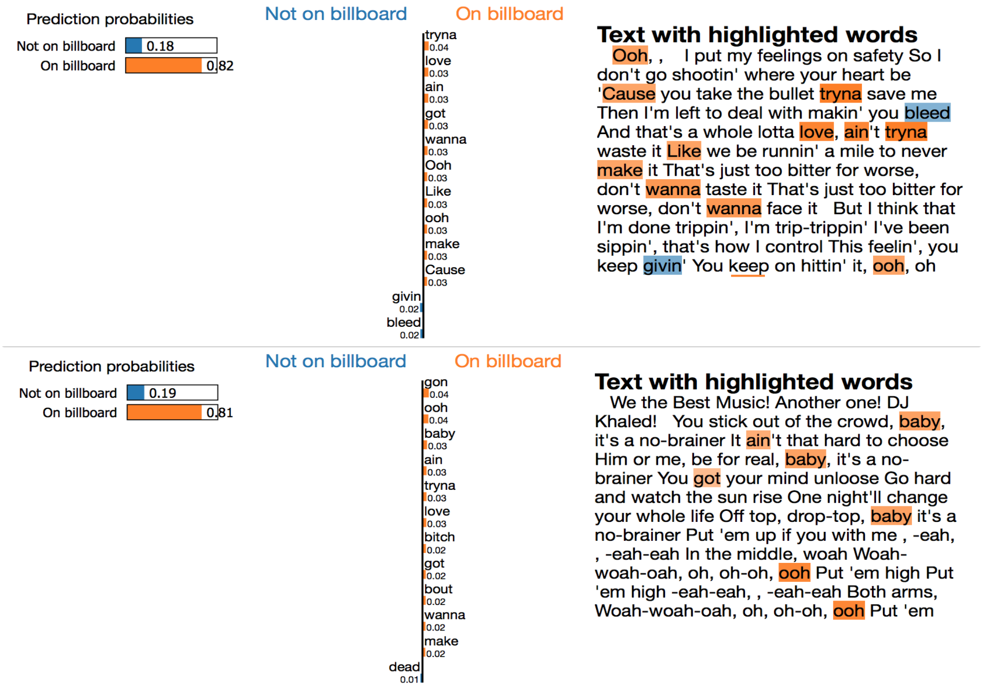
Figure 10: Audio Features ROC, Importance Level, and Correlation Matrix

4.4 LIME

In order to see the reasons behind the machine learning predictions, we deploy the package LIME(Local Interpretable Model-agnostic Explanations), an algorithm that can explain the predictions of any classifier or regressor in a faithful way, by approximating it locally with an interpretable model. It tells us the reasons why a certain machine learning model reaches the prediction result as opposed to seeing it as a black box. [4] We applied LIME to Logistic Regression Model in Section 4.1 and would like to share a few exmples.

In this case, LIME actually bridges the classifier model from our pure lyrics analysis to the model based on audio feature, as well as previous sentiment analysis. The following are respectively two hip hop songs that had been on the billboard for a while: Trip by Ella Mai and No Brainer by DJ Khaled featuring Justin Bieber.

Notice that the words that contribute to the overall positive predictions have more positive or proactive connotations in them. "Ooh" sound treated as instrumental (see the definition of "instrumentalness") also contributes to the positive prediction. This somehow intertwines the lyrics and audio features models together by relating the lyrics analysis to the two features "Instrumentalness" and "Valence".



Text with highlighted words

Ooh, , I put my feelings on safety So I don't go shootin' where your heart be 'Cause you take the bullet tryna save me Then I'm left to deal with makin' you bleed And that's a whole lotta love, ain't tryna waste it Like we be runnin' a mile to never make it That's just too bitter for worse, don't wanna taste it That's just too bitter for worse, don't wanna face it But I think that I'm done trippin', I'm trip-trippin' I've been sippin', that's how I control This feelin', you keep givin' You keep on hittin' it, ooh, oh

Text with highlighted words

We the Best Music! Another one! DJ Khaled! You stick out of the crowd, baby, it's a no-brainer It ain't that hard to choose Him or me, be for real, baby, it's a no-brainer You got your mind unloose Go hard and watch the sun rise One night'll change your whole life Off top, drop-top, baby it's a no-brainer Put 'em up if you with me , -eah, , -eah-eah In the middle, woah Woah-woah-oah, oh, oh-oh, ooh Put 'em high Put 'em high -eah-eah, , -eah-eah Both arms, Woah-woah-oah, oh, oh-oh, ooh Put 'em

Figure 11: LIME examples: Trip and No Brainer

5 Deployment

5.1 Concrete Deployment

We set up a website by Django framework to demo our project with our best performing baseline model. Here is the link. Songwriters can input the lyrics and expect an outcome that tells them if the song will become popular and which specific words should be retained in the lyrics (by LIME).

As was discussed before, one problem with our model is that we separated the lyrics and audio features of the hip hop music in our models, and so far haven't found an easy way of combing the two parts. We would say this is a limitation of our project. Nevertheless, the lyrics of music does reflect the overall vibe, tempo and melody of a song, so we do not think the problem greatly precludes the practical application of our project. The model will still function as a helpful tool for musicians or producers who would like see how the their work will turn out before release it or making minor changes to cater to the public's taste.

5.2 Code

All of our code is made available open-source in our Github.

6 References

[1] Thompson, Derek.*Hit Makers: How Things Become Popular*.Penguin Books, 2018.

[2] "Topic Modeling in Python with Gensim." *Machine Learning Plus*, 4 Dec. 2018, www.machinelearningplus.com/nlp/topic-modeling-gensim-python/.

[3] Ng, Andrew Y, and Michael I Jordan.*On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes*,4 Dec.2018,ai. stanford.edu/ ang/papers/nips01-discriminativegenerative.pdf.

[4] Ribeiro, Marco Tulio, et al.*"Why Should I Trust You?" Explaining the Predictions of Any Classifier*.5 Dec. 2018, www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf.

7 Appendix-Contributions

Bofei Zhang: scraped data from Spotify; merged and cleaned dataset;selected top features and set up models; parameter tuning; tested models in LIME; published project DEMO

Shaoling Chen: crawled lyrics from Genius and extracted data from Billboard;merged and cleaned data; descriptive statistics; sentiment analysis; learned a word embedding and tested it in a neural network model

Ziyu Lei: topic modelling by LDA; helped with lyrics crawling from Genius; helped with model selection;helped with final write-up drafting

Zian Chen: did research in Topic Modelling and LIME; embedded data visualization;helped with lyrics crawling;drafted final write-up