Using Machine Learning Tools to Classify Tweets of Popular Music Artists.

Joey Chao[*]
[*]Corresponding Author

Introduction

One of the most common ways for music artists to promote their content, causes, and brands is by using Twitter. They also use the platform to voice their opinions and interact with their fans. Artists typically use unique diction, slang, and structure that distinguish them from other artists. We can use this information to help project artist's popularity as well as to help identify fraudulent activity.

The goal of this project was to collect the most recent tweets from 10 popular mainstream music artists to build models to predict the author of tweets. In addition, this project aimed to create an unsupervised model that would be able to relate words and topics unique to the music industry and pop culture. I used the bag of words (BoW) and TF-IDF methods with supervised learning classifiers to predict authors of tweets. In addition, I used Word2Vec method to train a model on semantic meanings of words and topics.

I started the project by running Logistic Regression, XGBoost, and LGBM Classifier on bag of words styled features. Bag of words is a common natural language processing (NLP) technique that counts the frequency of words used in the collection of documents and generates features based on each word. The second approach I used to generate features from the tweets was to use term frequency-inverse document frequency (TF-IDF) to create feature vectors. TF-IDF, based on the bag of words method, is a popular method that first counts word frequency in a document then diminishes the weight of extremely common words in the document while increasing the weight of rare words. TF-IDF is commonly used as a NLP tool to classify text.

In addition, I trained a Word2Vec model to gain insight into the semantic meanings of the words in the collection of tweets. Word2Vec is a method that uses a neural network to identify semantic meaning of words. Word2Vec works by creating a large vector space and assigning unique words to vectors in that space. Words that appear near each other are assumed to have similar contexts and therefore the corresponding vectors of the words are placed near each other. With large corpuses, Word2Vec is able to assign very accurate meanings to words. Many large websites provide Application Program Interfaces (APIs) that help facilitate data collection. In Twitter's case, I used the Tweepy API to collect the most recent tweets by each of the artists I want to classify. APIs are tools and communication protocols that allow developers to access data and code needed to build software and collect data.

Methods

Collecting Data

Twitter provides access to tweets on the website through the use of their API. For this project, I used the Tweepy to access and collect the most recent tweets of 10 artists. One limitation imposed by the API was that only the most recent 3200 tweets could be accessed. This resulted in a relatively small dataset that may be challenging to implement in Word2Vec, but we could add additional corpuses help train the model. After saving the collection of tweets csv, I was ready to clean the data. After removing duplicate and empty tweets, I got between 1900 and 2800 tweets per artist (figure 1). After concatenating the data, I standardized the tweets by removing special characters, emojis, and url links. The remaining text was then tokenized and lemmatized in preparation for modelling.

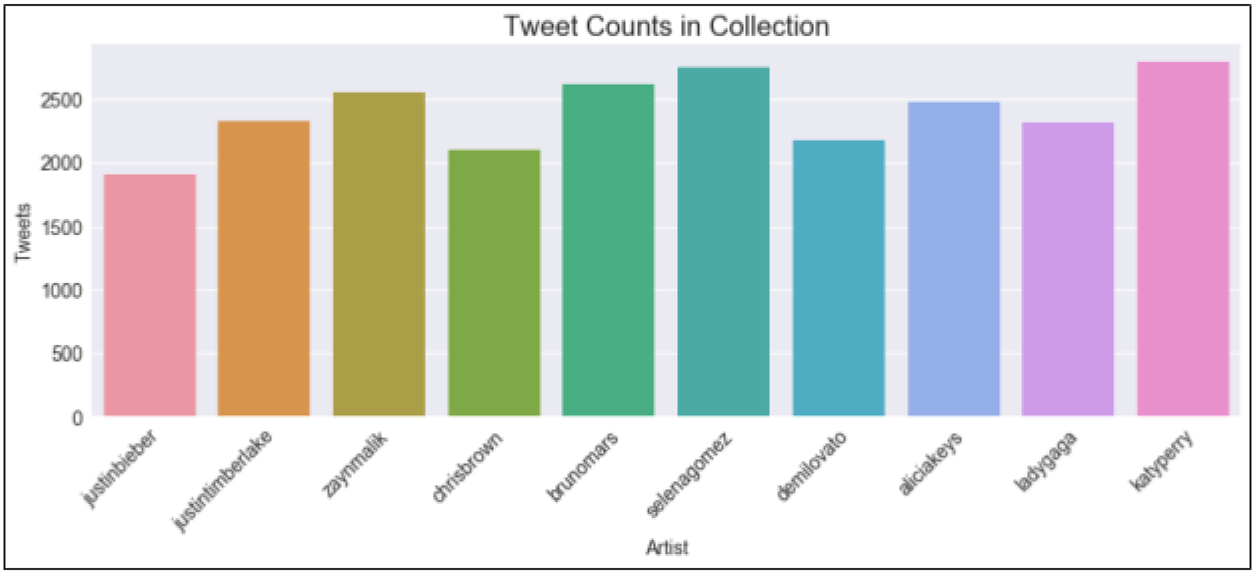


Figure 1. Final tweet count per artist.

1. Bag of Words

I used the CountVectorizer to create features based on the frequencies of unique words in the collection of tweets. The logistic regression classifier trained on the BoW features resulted in an accuracy score of 55.7%. I attempted to improve this score by adding tweet level features: word counts, character counts, punctuation counts, hash counts, and mention counts. These features were then scaled using StandardScaler and added to the original feature set. The model performances (figures 3, 4) improved significantly to 60.0%.

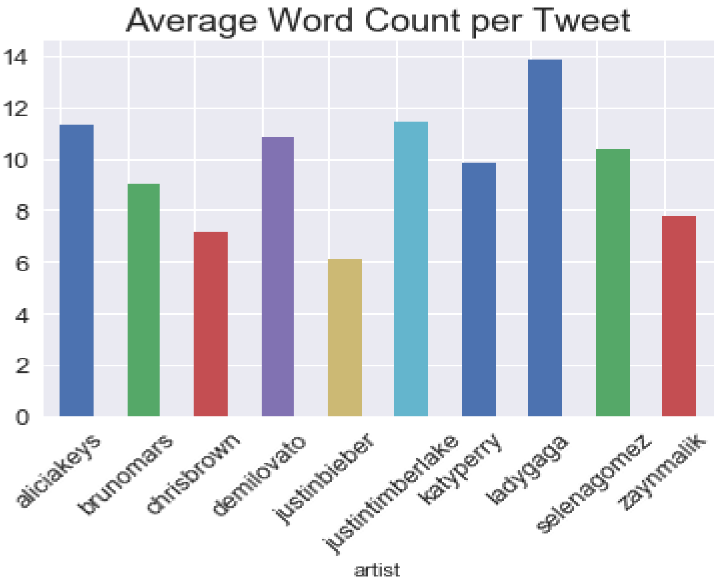


Figure 2. This graph shows the average word count per tweet for each of the 10 artists.

2. TF-IDF

I followed my bag of words features by transforming the original features to 6913 TF-IDF features. Using the logistic regression classifier, I got an accuracy score of 56.0% (figure 3). The score using TF-IDF features was much lower compared to the bag of words model. This was likely caused by the removal of uncommon words in tweets that were important in distinguishing the author. To resolve this issue, I changed the allowed minimum document frequency to allow for more features; this resulted in a slight accuracy boost to 58.9% (figure 3), but increased training time fivefold. To attempt to create a more efficient model, I used SVD to reduce the amount of features to 6000. Although 6000 features was calculated to explain 97% of the variance, the trained models were only 51% accurate (figure 3).

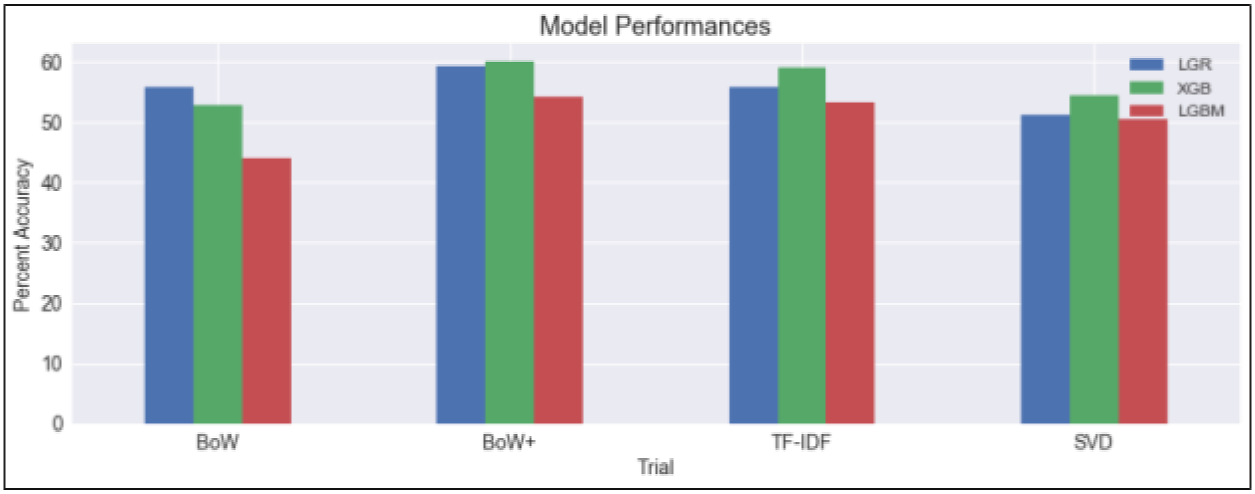
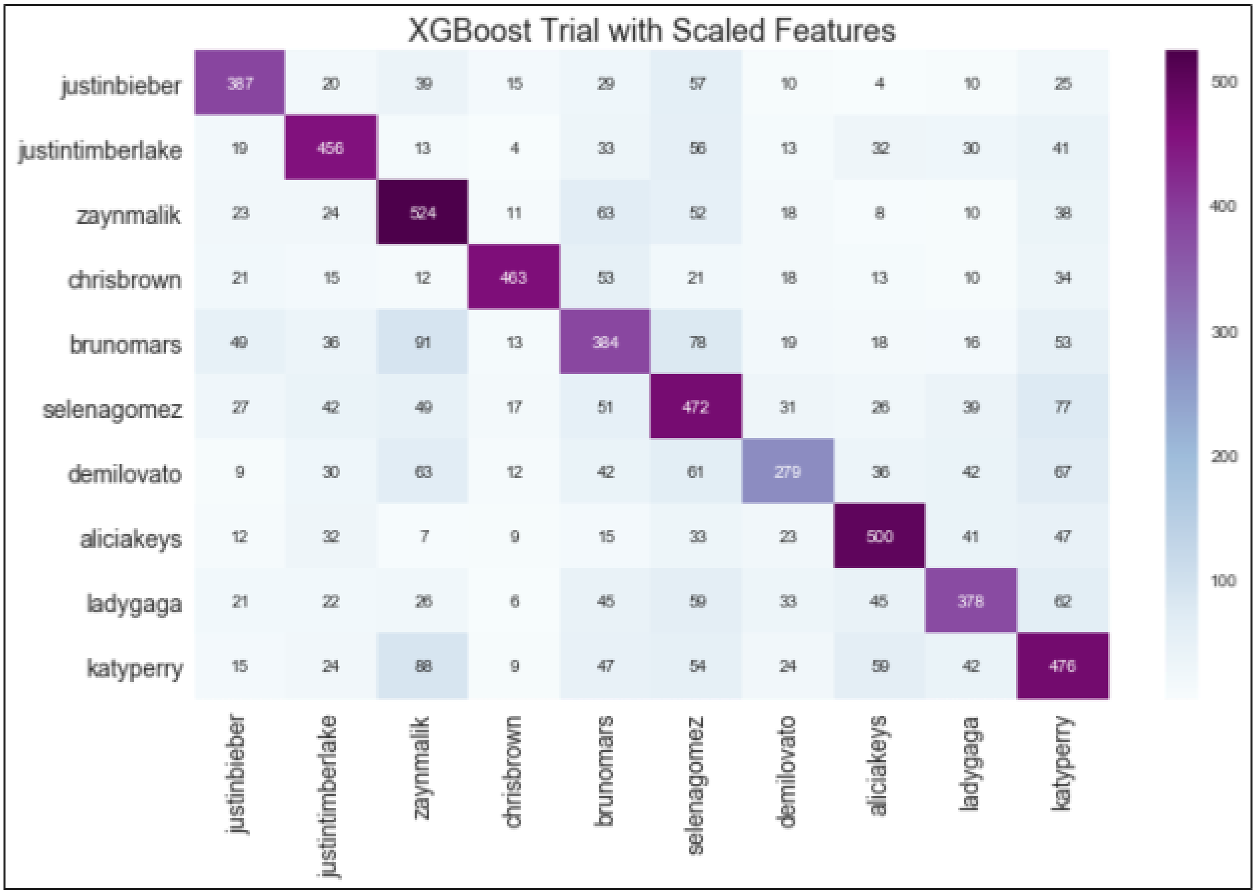


Figure 3. Model performances for trials: Bag of Words, Bag of Words + tweet level features, TF-IDF features, and SVD reduced features.



XGBoost Trial with Scaled Features

	justinbieber	justintimberlake	zaynmalik	chrisbrown	brunomars	selenagomez	demilovato	aliciakeys	ladygaga	katyperry
justinbieber	387	20	39	15	29	57	10	4	10	25
justintimberlake	19	456	13	4	33	56	13	32	30	41
zaynmalik	23	24	524	11	63	52	18	8	10	38
chrisbrown	21	15	12	463	53	21	18	13	10	34
brunomars	49	36	91	13	384	78	19	18	16	53
selenagomez	27	42	49	17	51	472	31	26	39	77
demilovato	9	30	63	12	42	61	279	36	42	67
aliciakeys	12	32	7	9	15	33	23	500	41	47
ladygaga	21	22	26	6	45	59	33	45	378	62
katyperry	15	24	88	9	47	54	24	59	42	476

Figure 4. Confusion matrix of XGBClassifier on the Bag of Words and tweet level features.

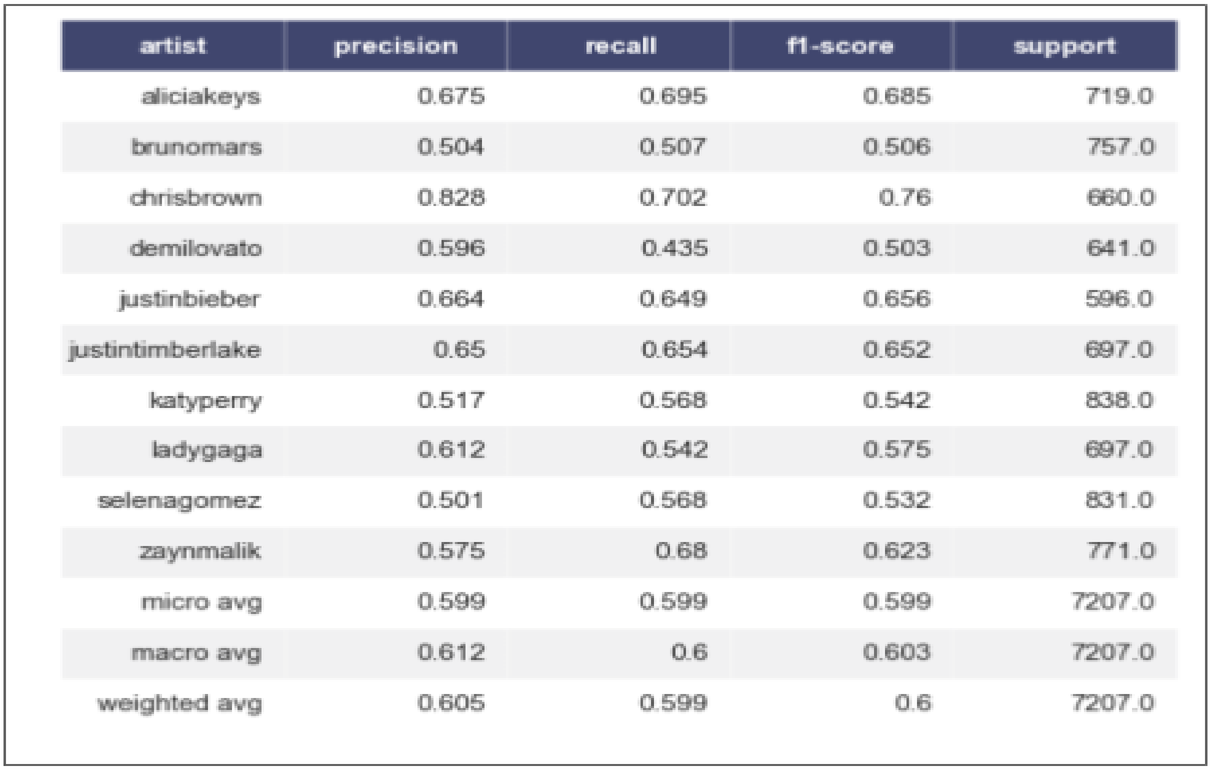
3. Word2Vec

I joined all 24,021 tweets into a single document and deployed the gensim implementation of Word2Vec. The resultant collection of words was relatively small at 499,007 words. After stop words were removed, I was left with 248,719 tokens in 14336 sentences. The corpus was relatively small, but was still able to give interesting predictions (figure 7).

Results and Discussion

Bag of Words and TF-IDF

I was able to achieve a 60.0% accuracy score in classifying 10 artists by training XGBoost on the bag of words features. Logistic Regression resulted in a slightly lower score at 59.2% while LightGBM produced the poorest results at 54.3% (figure 3). The precision scores ranged from 50% to 83% per artist and the recall scores ranged from 44% to 70% per artist (figure 5). Scores for the TF-IDF extracted features were similar but slightly lower when compared to the Bag of Words features (figure 3). Chris Brown's tweets were consistently the most accurately predicted. The difference in accuracy scores between each artist is likely due to the language they use in their tweets. Tweets by some artists like Chris Brown often link to other websites or are exclusively emojis. This makes his tweets unique compared to the other artists in the collection which helps the model make better predictions of his tweets. Justin Timberlake distinguishes his tweets by adding fan engaging hashtags such as *#teamJT*. The tweets also contain slightly different slang from different pop subcultures that may act as markers for the models to use.



artist	precision	recall	f1-score	support
aliciakeys	0.675	0.695	0.685	719.0
brunomars	0.504	0.507	0.506	757.0
chrisbrown	0.828	0.702	0.76	660.0
demilovato	0.596	0.435	0.503	641.0
justinbieber	0.664	0.649	0.656	596.0
justintimberlake	0.65	0.654	0.652	697.0
katyperry	0.517	0.568	0.542	838.0
ladygaga	0.612	0.542	0.575	697.0
selenagomez	0.501	0.568	0.532	831.0
zaynmalik	0.575	0.68	0.623	771.0
micro avg	0.599	0.599	0.599	7207.0
macro avg	0.612	0.6	0.603	7207.0
weighted avg	0.605	0.599	0.6	7207.0

Figure 5. Classification report of XGBoost model on the Bag of Words features.

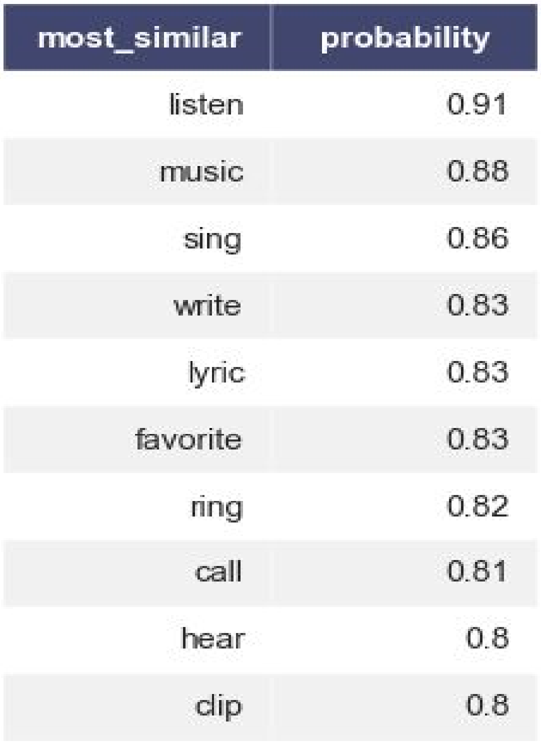
Word2Vec

Word2Vec implementation resulted in a limited vocabulary with mixed success. The model is able to detect similarities in some words, but there is also noise and inconsistencies in some of the predictions. Spelling (slang and purposely misspelled words) and word structure (words that used to be hashtags do not have spaces) were particularly erroneous (figure 6, 7). Figure 6 shows words predicted by the model similar to the input word “song”; words on this list like “listen”, “sing”, “write”, and “hear” are all plausible activities that correspond to “song”. The Word2Vec model seems to perform better on simple words. The model runs into difficulty when trying to make predictions on complex topics. Spelling (slang and purposely misspelled words) and word structure (words that used to be hashtags do not have spaces) were particularly erroneous (figure 7). Predictions in figure 7, “theexperience”, “jttour”, and “iheartradio”, were likely hashtags that were commonly used in sentences near “teamjt”. Hashtags in tweets are a challenge because they provide semantic value, but are also hard to deal with because they often consist of multiple words without spacing. Future work to parse these hashtags may help to retain the semantic value while creating a more standard document.

Conclusion

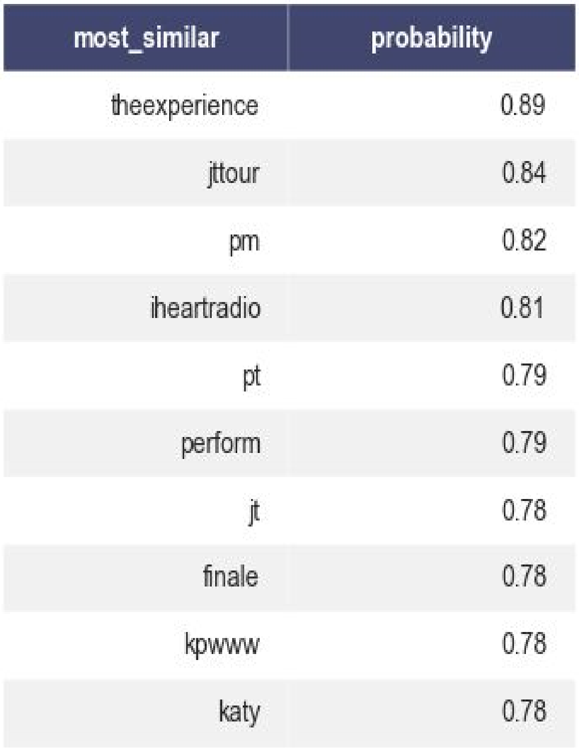
Modelling user tweets can help to identify fraudulent activity or identify trending topics. Of all the techniques I used, bag of words seemed to have the best performance likely due to the relatively small data size. A possible extension of this project is to stream data from twitter over a set period of time (between a month and a couple of years) for multiple artists, and using the expanded corpus to create a Word2Vec or Sense2Vec model that is sensitive to slang as well as popular music topics. This model has the potential to provide semantic model targeted at the music industry.

Figure 6. Word2Vec similarity scores for 'song'.



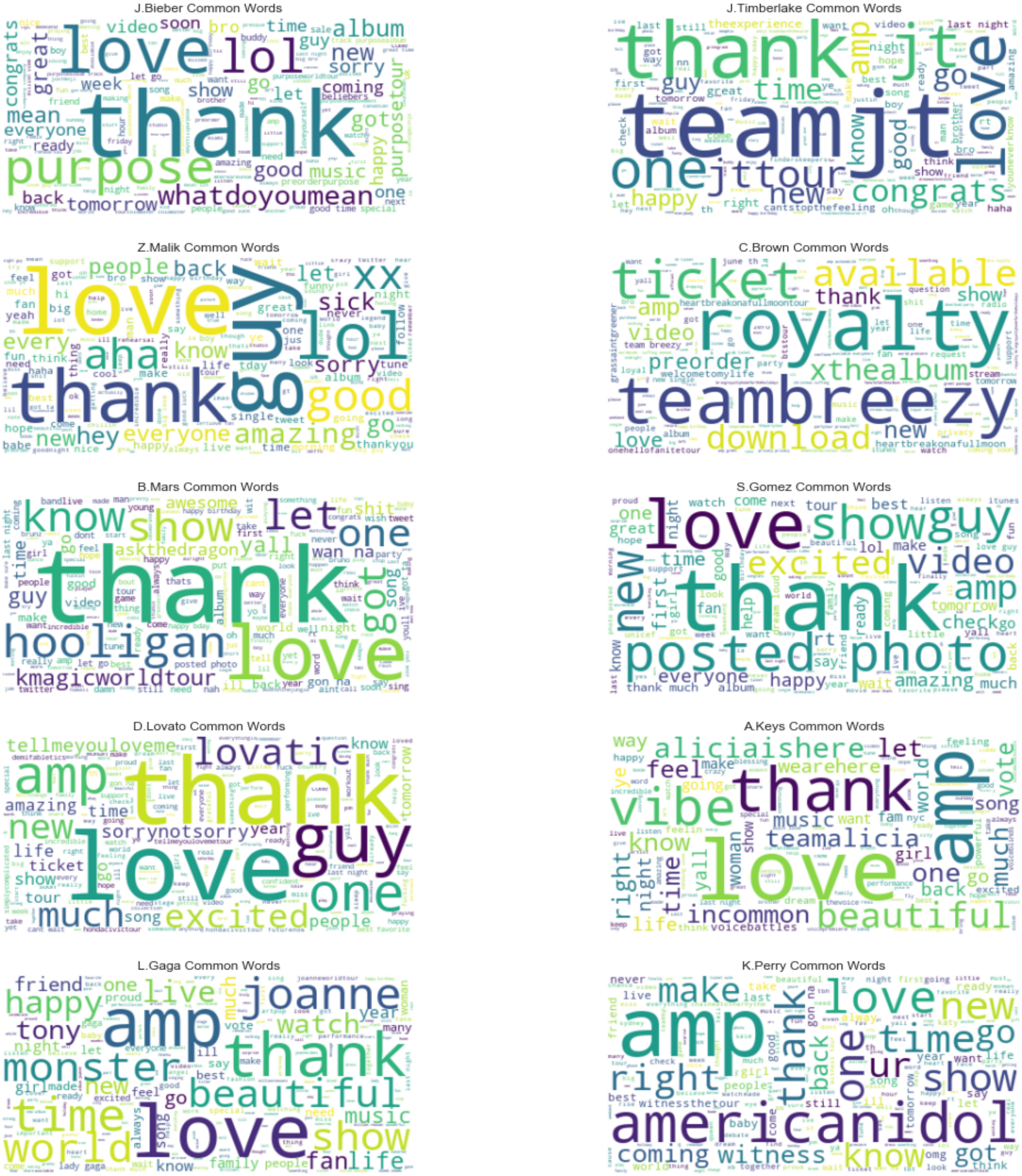
most_similar	probability
listen	0.91
music	0.88
sing	0.86
write	0.83
lyric	0.83
favorite	0.83
ring	0.82
call	0.81
hear	0.8
clip	0.8

Figure 7. Word2Vec similarity scores for 'teamjt'



most_similar	probability
theexperience	0.89
jttour	0.84
pm	0.82
iheartradio	0.81
pt	0.79
perform	0.79
jt	0.78
finale	0.78
kpwww	0.78
katy	0.78

Figure 8. Word Cloud image showing the most common words used by artist.



J.Bieber Common Words

J.Timberlake Common Words

Z.Malik Common Words

C.Brown Common Words

B.Mars Common Words

S.Gomez Common Words

D.Lovato Common Words

A.Keys Common Words

L.Gaga Common Words

K.Perry Common Words