Using Social Media Metrics to Predict Artist and Album Success

Pedro Garzon
Stanford University
pgarzon@stanford.edu

Vinson Luo
Stanford University
vluo@stanford.edu

Reynis Vazquez
Stanford University
reynis@stanford.edu

I. Introduction

Today's music industry is influenced by many factors that simply did not exist twenty years ago—current artists are capable of reaching out to potential fans using forms of online media. They are no longer limited in promoting their music by physical word of mouth or the advertising muscle of a record label. Instead, both artist who are just starting out and who had multiple album hits are using social media to build a presence, brand, and promote their current and upcoming music releases. However, the mediums used to consume music have changed as well. Music can be consumed instantly via online purchase or by streaming. More of our interactions with music itself are online rather than on music store shelves or the radio. This makes artists more independent of having to physically push albums.

Given metrics on social media platforms, we'd like to predict the success of an artist's upcoming release. Essentially, helping artists who can produce quality music benefits the entire industry, both on the consuming and producing side. Identifying more of these high potential artist early on can allow labels to give more artists the opportunities and resources to succeed sooner. With more resources, an artist gains more recognition and acclaim, more fans are generated, and the label maximizes their profit.

We'll be measuring success of an upcoming album release by looking both at an artist's track sales along with an artist's iTunes album sales one week after an album release. In order to make our predictions of album success, we'll be looking at social media metrics two weeks prior to an upcoming release. These social media metrics include Facebook likes, Twitter followers, LastFM plays, Myspace friends, and a few others that are part of our dataset provided by Next Big Sound. Social media metrics like these are generated by real people being engaged with the music content. Thus, it should follow that with these metrics as features, we can use machine learning methods to predict an album's success. In this paper, we'll use linear regression, naive bayes, and support vector regression to see how well we can predict an album's success in terms of both total iTunes tracks downloads and album downloads one week after an album release.

II. Related Work

Previous work on using the Internet to predict album success has yielded some initially unintuitive and contradictory results. An early study of finding the relationship between user generated Internet content and albums sales looked at blog posts and album reviews to see if they correlated with album sales. The results were positive. Essentially, the more legitimate blogs and album reviews about a certain album, the more buzz it had in the Internet community, and so the more physical album sales were sold due to the popularity. In addition, it was found that only $\frac{1}{6}$ of ”high chatter” albums were debut albums, which suggests that the biggest factor in a successful release is having had at least one prior successful release. This study used several linear regression varying in which independent variables were used such as number of blog reviews, sales rankings, and MySpace friends. p-values were analyzed to make its analysis. The results suggest that high Internet content generally happens when an artist has already had a quality release. [1]

However, a similar study conducted a few years later had opposite results. They found that buzz generated by blogs tended to have no correlation with increasing album sales. More interestingly, it was seen that having a large amount of hype on the Internet for a modest artist results in a negative impact in track sales. Rather than high buzz allowing for an artist's music to be more well known and thus have more people buying it, it turns out that the higher amount of recognition results in lower sales at the track level. This isn't the case for an already mainstream artist, however. This study attributes this finding to blogs talking about small artists that tend to post ways to engage with individual tracks of an artist via mp3 download links or links to free streaming. Thus, listeners themselves promote artists in ways that make the music available for free. The study used panel vector autoregression (PVAR) to look at the bidirectional relationship between the dependent and independent variables. This allowed a way to see if they might be affecting each other and allowed for a more comprehensive analysis of how Internet media affects album and track sales. This study was limited in that it's measure of buzz was limited to blog sites and not social network sites like Facebook and Myspace. [2]

Another study took a different approach by only directly using one social music site's data: Last.fm. This study created features at the artist and track level by processing thousands of listeners' Lastfm charts. Features such as total amount of listens on a tracks and total amount of listeners was generated via scrapping Lastfm. Using the generated features, the researchers attempted to predict the Billboards rankings of the tracks. They used Support Vector Machines, Naive

Bayes, Bayesian Networks and Decision Trees and got the best accuracy result of 81.31% in predicting track placement in Billboards. [3] This study shows that looking at a social network's metrics can give some indicating of a track sales. It then makes sense that a high amount of Last.Fm consumption of a track correlates with track purchases since it models daily consumption on users' devices. The methods used were standard regression models common in machine learning.

From 2012, part of the year our dataset will be from, we see that there have been almost 94 billion plays and 17 billion profile views across different social sites as calculated by Next Big Sound [4]. This tells us that there's is clearly an enormous amount of engagement with artists and their product online. Next Big Sound analyzed the metrics that they collect on artists and concluded which social media metrics contribute the most to an album's sales. Using a Granger causality test, which looks at whether one timeseries of data is able to forecast another, Next Big Sound found that Facebook likes, Wikipedia views, Vevo plays, YouTube plays, and SoundCloud plays were most significant in forcasting album sales [5]. This suggest that it should be possible to use machine learning methods on such features to predict album sales at least.

III. Dataset and Features

The dataset that we used to test build our models was Next Big Sound's Challenge dataset, a set containing social media metrics along with YouTube play counts and iTunes sales data for anonymized artists over the course of roughly two years. There was data available for 1818 artists over a total of 909457 artist-days in the years 2011 and 2012.

For each artist, a variety of different social media metrics were available for Facebook, Instagram, Twitter, Last.fm, MySpace, and SoundCloud in addition to several other music sites. Notably, the date of album releases by artists was also provided, allowing us to narrow our focus to the prediction of short term album success (measured by the number of sales made by an artist in the week following an album release) based on social media metrics.

Much of the data, however, was filled with NaN's, indicating that artists did not have profiles in the various social media services. The median percentage of days for which a metric was unavailable was a high 79.7%, with only 17 of the 100 fields containing non-NaN data for over 50% of all artist days.

Because values for most of the metrics spanned several orders of magnitude (very popular artists, for example, could have hundreds of times more tracks sold than average artists), we decided to use the log of most metrics in place of their actual values in our analysis.

We initially settled upon using the number of iTunes tracks sold in the week following an album release as our y variable for regression, as it was the most widely available metric related to artist success (available for 80% of the data). However, we quickly discovered that, for many artists, album releases did little to change their sales counts from what they were prior to these album releases. As a result, we decided to also look at predicting the change in track sales following an album release (measured as $\Delta_{tracks} = log(sales\_after/sales\_before)$) as a possible measure of *album* and not *artist* success. **Figure 1** shows the distribution of the Δ_{tracks} variable.

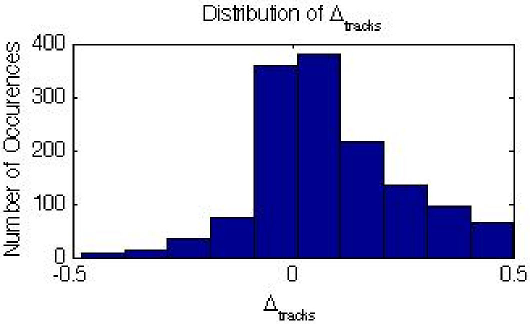

Fig. 1. Distribution of Δ_{tracks}

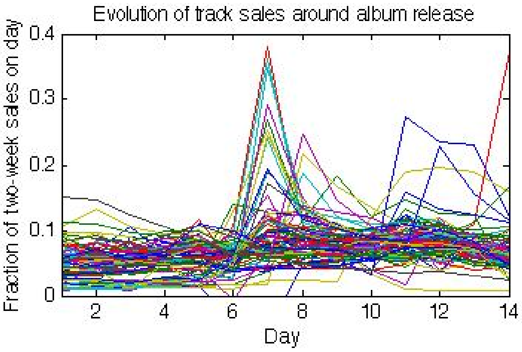

Fig. 2. Evolution of track sales around album release

This resulted in a total of 1543 album releases that could be used as examples for both training and testing. 100 random samples of the evolution of track sales per day for an artist around an album release are shown in the **Figure 2**. Here, day 7 (the day corresponding to peaks in several of the sales paths) represents the day that an album was released.

To deal with the presence of NaN's inside the data, we used different methods that varied based on the learning algorithms we used. For some of our models, we also imposed the additional restriction that artist data for the 7 days prior to an album release also be available so that we could use more advanced time-based predictors besides the metrics on the day of a release.

One of the most relevant time-based features was the "acceleration" a_t of an artist's track sales one week prior to release, computed as:

$$a_t = \frac{s_{t-1} - s_{t-7}}{s_{t-1} + s_{t-7} + 1}$$

where s_t is the total number of iTunes track sales on artist-day t. This acceleration metric gives a rough sense of how quickly track sales are either increasing or decreasing, scaled so that its value always lies between -1 and 1.

IV. Methods

A. Linear Regression

The first form of regression we tried on the data was a simple ordinary least squares linear regression. We were looking to predict a real valued metric based on a vector of features at every album release, so using least squares was a reasonable first approach. Because there is no easy way to deal with NaN's in a linear regression, for every set of features we tried we had to restrict our data to only those artist-days for which all features had actual values.

We then constructed a predictor matrix X with each row corresponding to values of the predictors on an artist-days selected by the criteria above. A column of ones was added at the end of the X vector to include an intercept term in the linear regression. y simply consisted of the values of Δ_{tracks} at the corresponding days. Predictors we tried using included Facebook likes, Twitter followers, and iTunes track sales acceleration.

The least squares regression model seeks to fit a parameter θ such that predictions of a responding variable y given an observed vector x are given by

$$h_\theta(x) = \theta^T x$$

Formally, we minimize the sum of squared differences between the values of $h_\theta(x^{(i)})$ and $y^{(i)}\}$ given observations $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$. This is done by minimizing the cost function

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

In our formulation, this problem simplifies to minimizing $J(\theta) = \frac{1}{2}(X\theta - y)^2$ with respect to theta, yielding the optimal value $\hat{\theta}$ given by

$$\hat{\theta} = (X^T X)^{-1} X^T y$$

B. Naive Bayes

The Naive Bayes classifier provides a more explicit method of dealing with NaN's, but in order for the classifier to be used, both the predictors and responding variable have to first be discretized into separate classes.

Values for each metric were turned into discrete features by first shifting all values so that the *median* value of the metric was 0, and then bucketing the values based on the standard deviation σ of all non-NaN values. While it may seem odd that we decided to center data on the median rather than the mean, in practice using the median results in massive improvement due to the presence of extreme outliers in some metrics. The buckets b_i were assigned in the following manner: $(-Inf, -1.75\sigma) \mapsto b_1, (-1.75\sigma, -1.25\sigma) \mapsto b_2, \ldots (1.25\sigma, 1.75\sigma) \mapsto b_8, (1.75\sigma, Inf) \mapsto b_9, NaN \mapsto b_{10}$.

This allowed us to change each predictor $x^{(i)}$ and responder $y^{(i)}$ into a vector of multinomial variables (each of which has 10 possible values). If we interpret the discrete output bucket i of the classifier as predicting y to be the center c_i of their corresponding bucket, then we can also obtain a standard error for the Naive Bayes classifier by comparing this center to the actual value $y^{(i)}$ for that observation.

The Naive Bayes classifier itself is based on the strong assumption that all individual features x_i (in our case the values of the multinomial variables corresponding to each field) in observation x are conditionally independent given the corresponding class y. This allows us to simplify the conditional probability $p(x|y)$ of observation x occuring given class y into

$$p(x|y) = p(x_1, \ldots, x_n|y) = \prod_{i=1}^{n} p(x_i|y)$$

where all values of $p(x_i|y)$ can be much more efficiently computed and stored than values of $p(x_1, \ldots x_n|y)$. Using Bayes' Rule, we can now easily use these conditional probabilities $p(x|y)$ along with class probabilities $p(y = k)$ to make predictions:

$$p(y = k|x) = \frac{p(x|y = k)p(y = k)}{p(x)}$$

The best class fitting an observation x would then be given by $\arg\max_k p(y = k|x)$.

Maximum likelihood estimates for the multinomial variables in the Naive Bayes classifier are easily fit by simply tallying up the occurrences of $\{x_j^{(i)} = l \wedge y^{(i)} = k\}, i = 1, \ldots, m, j = 1, \ldots, n$ for $l, k = 1, \ldots, n_{categories}$, where n is the number of features for each $x^{(i)}$ and $n_{categories}$ is the number of discrete categories for both the predictors and responder (in our case 11). To assign reasonable nonzero probabilities to the occurrence of all events, we use Laplace smoothing, a technique equivalent to initializing the classifier with one occurrence of each $\{x_j^{(i)} = l \wedge y^{(i)} = k\}$ event.

It is true that the features in our data certainly violate the Naive Bayes assumption; Facebook likes and Twitter followers, for example, actually show very strong correlation despite having not much correlation with Δ_{tracks}. Practically speaking, however, Naive Bayes typically still works well even in situations where there may be correlation between input variables.

C. Support Vector Regression

ϵ-support vector regresssion allows us to find a function that tries to maintain within the boundary specified by ϵ, and minimize the cost of errors outside ϵ. This model also maps the features to a higher-dimensional space and uses a Kernel to easily compute the inner product of our feature mapping.

ϵ-support vector regression can be posed as the following optimization problem

$$min \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i + \xi_i^*$$

$$s.t. \quad \begin{cases} \mathrm{y}_i - \langle w, x_i\rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i\rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

We can then use a kernel and construct a Lagrange function to produce the dual optimization problem:

$$max \quad \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i - \alpha_i^*)K(x_i, x_j) \\ -\epsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*) \end{cases}$$

$$s.t. \quad \begin{cases} \sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases}$$

We can calculate $w^T x + b$ as the following (and calculate b by using the Karush Kuhn Tucker conditions):

$$w = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i$$

$$f(x) = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)K(x_i, x_j) + b$$

$$b = y_i - \langle w, x_i \rangle - \epsilon \text{ for } \alpha_i \in (0, C)$$

$$b = y_i - \langle w, x_i \rangle + \epsilon \text{ for } \alpha_i^* \in (0, C)$$

To account for the possible non-linear relation of the data, we used the radial basis function (RBF) kernel

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2)$$

where we chose $\gamma = \frac{1}{m}$.

V. Results

To evaluate the performance of each of our models, we evaluated their standard error in the prediction of Δ_{tracks} on a test set after being trained on a separate training set. Models that showed promise were subjected to more rigorous cross-validation tests. As an absolute measure of how effective each of the methods were, we compared the percentage of variance in Δ_{tracks} explained by a method.

A. Linear Regression

The linear regression approach we initially used was incapable of dealing with NaN's, so we had to narrow down our initial set of album releases to only the 996 releases that contained both artist Facebook and Twitter data.

It's important to note that by throwing out those artists who do not have both Facebook and Twitter accounts, we inherently bias the values of Δ_{tracks} that we look at. Artists that have both Facebook and Twitter accounts typically have better album success (as measured by Δ_{tracks}), indicating that there is definite information that can be gleaned from the fact that artists are missing a social media profile. For now, we set aside this difference and continue to focus on those artists that have both Facebook and Twitter data to see how these two predictors factor into an artist's change in track sales.

Our results match up with the somewhat counterintuitive results of [2] in that Facebook likes and Twitter followers were actually negatively correlated with album success as measured by Δ_{tracks}. The covariance matrix for Facebook likes, Twitter followers, and Δ_{tracks} is shown in **Table 1**.

The resulting linear predictor had a standard error of 0.3598, corresponding to a variance of 0.1294. The additional two features, then, explain 5% of the variance of the restricted dataset (0.1361), adding a small amount additional predictive power. Because this value alone was so low, we decided that it was not worth it to further pursue cross validation tests with these features alone.

TABLE I. Covariance Matrix for Facebook likes (FB), Twitter Followers (Tw), and Δ_{tracks}

	FB	Tw	Δ_{tracks}
FB	0.9565	0.7090	-0.0769
Tw	0.7090	1.0508	-0.0423
Δ_{tracks}	-0.0769	-0.0423	0.1361

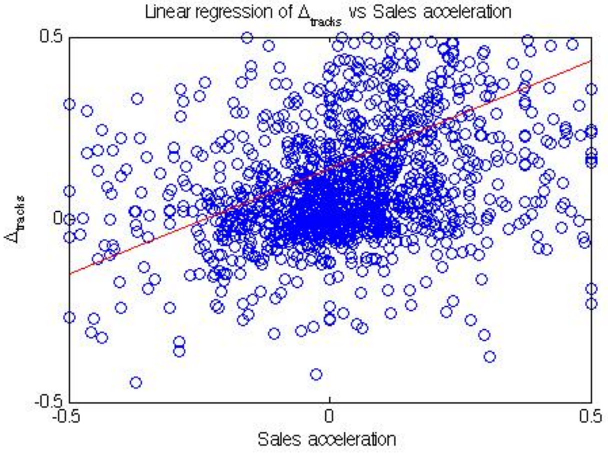

Fig. 3. Linear regression of sales acceleration and Δ_{tracks}

The use of a third feature, the acceleration of an artist's iTunes sales prior to an album release (computed as the percentage change in track sales per day over the week prior to a release), was what most improved the performance of the linear regression model. This feature had a high 0.456 correlation with Δ_{tracks} and alone was capable of explaining 21% of the variance of the set of 1543 test examples for which this metric was available (0.1297), attaining a standard error of 0.3205. A plot of acceleration vs Δ_{tracks} along with the corresponding regression line is shown in **Figure 3**.

To verify that this performance would hold even using out of sample data, we utilized a variant of k-fold cross validation that averaged the out of sample prediction error of this linear predictor over 1000 trials, with each trial randomly selecting 70% of the data for training and 30% of the data for testing.

The linear regression model using accelerations only performed well on this form of cross-validation. We computed the average standard error of the linear classifier to be 0.3208, corresponding to a variance of 0.1029, a significant decrease from the 0.1297 variance of Δ_{tracks} in all examples for which a_t was available.

Using both Facebook and Twitter metrics with artist iTunes acceleration did improve the average standard error of the classifier to 0.3203 under the same cross-validation scheme, corresponding to 25% variance explained (the smaller datset also had a higher variance of 0.1370). However, because the use of Facebook and Twitter metrics significantly restricts the number of albums that can be classified (1543 to 984), we decided that it was not worth including these metrics despite the slight improvement they produced in variance explained.

A summary of the results for linear regression is shown in **Table 2**.

TABLE II. Average Standard Error in OLS Regression of Δ_{tracks}

Predictors	Num examples	Error	Var. of Δ_{tracks}	% Var. explained
FB + Tw*	996	0.3598	0.1361	5%
a_t	1543	0.3205	0.1297	21%
All three	984	0.3203	0.1370	25%

* Not cross-validated

TABLE III. Average Standard Error in Naive Bayes Regression of Δ_{tracks}

Predictor Set	Average Standard Error	% Var. explained
1	0.332	15%
2	0.401	-23%*
3	0.331	15%
4	0.332	15%
5	0.331	16%

* Introduced additional variance (worse predictor than mean)

B. Naive Bayes

The Naive Bayes classifier was able to explicitly deal with NaN's, allowing it to work with all 1543 albums that had iTunes sales data. This capability also allowed us to easily test any feature we wanted on the resulting classifier.

We tried running the Naive Bayes classifier to predict Δ_{tracks} using several different sets of predictors. For each set, the average estimated standard error was computed using the same cross validation technique as linear regression, except this time 90% of the data in each of 1000 iterations was devoted to training and 10% was devoted to testing. This was designed to ensure that there was a sufficient amount of training data for the Naive Bayes classifier, which requires more data points (given the number of features it deals with) than linear regression.

Table 3 displays the estimated standard error of the Naive Bayes classifier using cross-validation. Set 1 contained just iTunes sales accelerations; Set 2 contained accelerations and Facebook likes; Set 3 contained accelerations and Last.fm plays; Set 4 contained accelerations and Twitter followers; Set 5 contained accelerations, Last.fm plays, Twitter followers, and Twitter tweets.

It's hard to tell from this data alone whether there exists a set of features would allow the Naive Bayes classifier to perform better than the 21% variance explained statistic derived from linear regression using accelerations. However, the fact that the widely available metrics of Facebook likes, Twitter followers, and Last.fm plays all did little to improve the error of the final model (Facebook likes actually hurt the model) doesn't bode well for classifiers using larger sets of predictors. Due to time limitations, we decided not to pursue further variations of the Naive Bayes model, though techniques such as forward search could definitely be used to optimize the set of predictors used.

C. Support Vector Regression

Using a radial basis function kernel and support vector regression, we mapped our Facebook and Twitter metrics features to a higher-dimension to account for a non-linear relationship within our features. This produced a standard error of 0.6128, producing a regression significantly worse than linear regression.

Adding the five features with the least amount of nan-values (Facebook page likes, Twitter followers, MySpace friends, MySpace followers, and MySpace profile views) to our regression caused a slight decrease in the standard error (0.5521). Despite the improvements, however, linear regression continued to be a better model.

To mitigate the problem of overfitting, we reduced the number of features down to three but this time, using Facebook likes, LastFM listeners and RadioWave Sirius Impressions. We chose these features based on [2] indicating that a bigger indicator of track sales is radio plays over social media buzz. However, since this restricted our data to 24 training examples, we restricted our training examples to just Facebook likes and RadioWave impressions but interpolated the missing NaN values of LastFM listeners using k-nearest neighbor imputation. This produced a standard error of 0.2384, an increase over our previous models yet still not a great model considering that this significantly reduced our training set to 139.

In trying to overcome initial problems of overfitting, we found that modifying the parameters by reducing the cost or increasing the range of "no-cost" errors (i.e. increasing ϵ) had either no effect or adverse effects on the testing error.

VI. Conclusion

In the end, the original linear regression model with only sales acceleration as a predictor was the best at predicting values of Δ_{tracks} across a large range of inputs, outclassing the discretized Naive Bayes regression and the SVR models on the full dataset. However, the use of SVR's did show promise for albums that had specific combinations of data available—in this case, the combination of Facebook Likes, LastFm listeners, and RadioWave Sirius impressions was able to achieve a lower standard error of 0.2384.

None of the models that could be applied to a large number of albums, however, were very accurate at predicting track growth, and those formulated purely on social media metrics performed noticeably worse than those that accounted for a history of iTunes sales data. This is most likely caused by the fact that there is high variance among social media metrics even given the same amount of track sales growth, a problem exacerbated by the fact that only a few metrics are typically available for each album.

Given more time, the Naive Bayes model could be further improved by implementing a cross-validated forward search to determine the optimal features to select for prediction. The Naive Bayes model is the most versatile of the three types of models, as it explicitly deals with missing data, and could potentially improve with an exhaustive search over all features (as well as meta-features that take into account the time series nature of the data). Robust imputation schemes that work with sparse data could also be tested to see if they allow for improved versions of linear regression and SVR.

Acknowledgment

Next Big Sound for the dataset.

REFERENCES

[1] Vasant Dhar, Elaine A. Chang, Does Chatter Matter? The Impact of User-Generated Content on Music Sales, Journal of Interactive Marketing, Volume 23, Issue 4, November 2009, Pages 300-307, ISSN 1094-9968, http://dx.doi.org/10.1016/j.intmar.2009.07.004. (http://www.sciencedirect.com/science/article/pii/S1094996809000723)

[2] Sanjeev Dewan and Jui Ramaprasad. 2014. Social media, traditional media, and music sales. MIS Q. 38, 1 (March 2014), 101-122.

[3] Huang, Ronghuai and Yang, Qiang and Pei, Jian and Gama, Joo and Meng, Xiaofeng and Li, Xue and Bischoff, Kerstin and Firan, ClaudiuS. and Georgescu, Mihai and Nejdl, Wolfgang and Paiu, Raluca. 2009. Social Knowledge-Driven Music Hit Prediction. Advanced Data Mining and Applications Volume 5678 (January 2009), 43-54.

[4] ”Next Big Sound - Analytics and Insights for the Music Industry.” Next Big Sound - Analytics and Insights for the Music Industry. 2010. Accessed December 6, 2015. https://www.nextbigsound.com/industryreport/2012/.

[5] Buli, Liv, and Victor Hu. ”Data Science And The Music Industry: What Social Media Has To Do With Record Sales.” Hypebot. 2012. Accessed December 6, 2015. http://www.hypebot.com/hypebot/2012/12/data-science-and-the-music-industry-what-social-media-has-to-do-with-record-sales.html.