Predicting the next big song hit, what is at the heart of a song?

Adams Marc marc.adams@epfl.ch
Prevel Paul paul.prevel@epfl.ch
Weiskopf Robin robin.weiskopf@epfl.ch

Abstract

Analysis and ML on the features of the Million song dataset to predict the next hit.

1 Introduction

The Million song dataset (Bertin-Mahieux et al., 2011) gives us data on popular songs between 1922 and 2011, with both metadata and processed data The goal of this work is to analyze this data and get a better understanding of the summer hit phenomenon and what makes a song popular. What features of a song makes it so catchy? Is it possible to predict if a song will be popular or not? We have a strong intuition it is, but supposedly answering that question demands a deep understanding of the social/cultural context as well as an analysis of the song. Social/cultural context would be for example: The popularity of the hippie movement and music was born as a contest to wars at that time. In the same spirit, early hip hop music was a contest to segregation.

In our case, we only have access to a dataset of music metadata and analysis. We will therefor explore the hypothesis that this relation can be found without the social/cultural aspect. Focusing on a small time line, and considering no important social/cultural event happens, we can easily imagine that this is the case. Why do all the most popular song of this year sound the same? Wouldn't it be because artists and record labels have become better at distilling the core of a catchy song?

We organized this report in the following manner: In part 2 we will describe the dataset we are using, in part 3 we will explore the data to find the range of values and stats, then in part 4 we will explain the methodology we used for our analysis, and finally we will discuss our results.

2 Dataset description

The Million Song Dataset is a collection of different datasets, musicbrainz.org and the.echonest.com, that have been compiled together for easier manipulations. It is composed both of metadata and processed analysis of the songs. In the pure metadata, we fine: the artist name, song name, location of where the song was composed, tags describing the style. In the processed analysis, algorithm have been execute on the original files that try to give meaningful information about them.

2.1 The metadata

Metadata is composed of the artist name, song name, location of the artist, tags given by musicbrainz.org, the weighted tags given by The Echo Nest, the song hotness given by the Echo Nest, the year of release according to Musicbrainz. In our analysis, we will consider that the hotness measure of the Echo Nest is the popularity of the songs. It is that value that we will try to predict given the features that we will select. Metadata like artist name, song name, and location of the artist will not be used, as we want to be able to predict the popularity of a song based only on the features of the song. On the contrary, tags of The Echo Nest and Musicbrainz are very interesting. They have values like: 'pop', 'hip hop', 'jazz', 'downtempo", 'pop rock'. These tags represent a human interpretation and feeling of the song, it is our only small insight of the cultural dimension of the problem.

2.2 The processed analysis

In the processed analysis, algorithms have been run to detect features: the key, tempo, energy, danceability, loudness, time signature, and where the bars start. The songs have also been divided into segments. A segment can be a chorus, a verse, or any meaningful part of the song. For each segment the dataset gives: the chroma pitch analysis, the timbre, the tatums. Key, tempo energy, danceability and time signatures are single

values that can easily be used in correlation, and machine learning. Segments of chroma pitches, timbre, tatum, bars on the other side are represented by matrices and vectors. Some preprocessing is required to extract the aspect of them we are interested in. We will discuss this further in the methodology.

3 Exploring the dataset

The first approach to analyzing the data, is getting to know it better. All the values are floats with values between 0 and 1. The distribution of song hotness is not uniform on the whole dataset, the histogram of those values is represented in figure 1. As we can see, not all values of hotness are represented as much. The mid range from 0.2 to 0.7 has many songs, but there's nearly no songs below 0.2, and above 0.7 we have less and less tuples. As we will see, it will be important to keep in mind this distribution in the methodology later, we will be using stratified sampling for the machine learning. About 120000 songs have no hotness value, so we will applying our algorithms on maximum 880000 tuples.

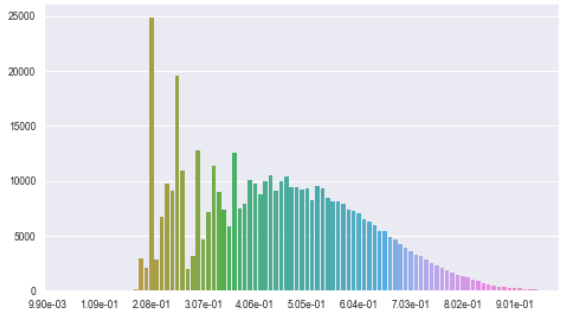

Figure 1: Histograms of the hotness values for the whole dataset

The distribution of the years is also important to keep in mind. As we discussed in the introduction, our analysis is sensitive to the time period, we always need to worry of the year distribution. As we can see in the histogram in figure 2, the years spread from 1922 to 2011. The histogram has an exponential style progression, so we will focus on songs in the period 2000 - 2011.

Unfortunately, we found that all the values of danceability and energy are 0 in the dataset. These analysis values are subjective of course, but as we consider the hotness values of The Echo Nest as true, and that these values also come from The Echo Nest, they would of been useful in our analysis later. For a full analysis of the values in the dataset, please refer to the *Milestone3.ipynb* notebook.

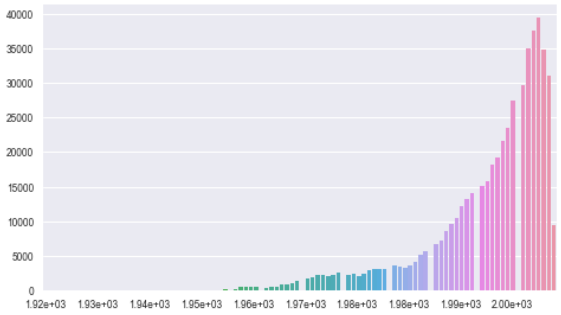

Figure 2: Histograms of song release year

4 Methodology

In this section we will describe our different approaches to solve the problem. The main idea is to try and find pattern between hotness and song features. Before we can try to predict the next hot song, we need to understand what makes a song good. The Correlations, Lyrics, Tags part are in preparation of the final part, the Machine Learning, which puts everything together.

4.1 Correlations

As a first mathematical measure, correlations naturally come up to mind. We applied Pearson's correlation between hotness and the simple metadata features: duration, key, tempo, loudness, time signature. The results are represented in figure 3. The correlation is only applicable to single values, applying it to the vectors and matrices would require some preprocessing, and an idea of what exactly would make sens to extract. Of course we didn't expect any results from key and time signature, the histogram of these values form a bell shape centered around the most common values i.e 4 (4/4, the standard time signature) and 120bpm (beats per minute). Increasing the key signature to get a 'hotter' song makes no sens. Increasing the tempo can make a song more lively, but increasing it too much just transform a song to a punk song!

What did come out though, is the correlation between loudness and hotness. To convince ourself of the results, we applied stratified sampling on the whole dataset, and plotted a scatter plot with hotness and loudness as the axis. The scatter plot is represented in figure 4. Each point in the plot is a song from the dataset. The red line is a least

square linear fitting of the points, expliciting the correlations between hotness and loudness for the points on the plot.

Our interpretation of this correlation is that a louder song is a song that has less calm and quiet parts in it. So increasing the loudness here would mean making a song more lively. Most of the popular songs are lively and have rhythms that make people want to dance.

For the stratified sampling, we divided the dataset in 10 partitions, where each partition has a range of hotness values, in that case $[0, 0.1), [0.1, 0.2), ..., [0.9, 1]$, and took 200 random points in each partition. The idea of doing stratified sampling was to get all hotness values represented equally in the plot, and that the plot would not be biased by the distribution of hotness values in the dataset.

duration	key	tempo	loudness	time signature
-0.019	-0.009	0.023	0.143	0.027

Figure 3: Correlation measures

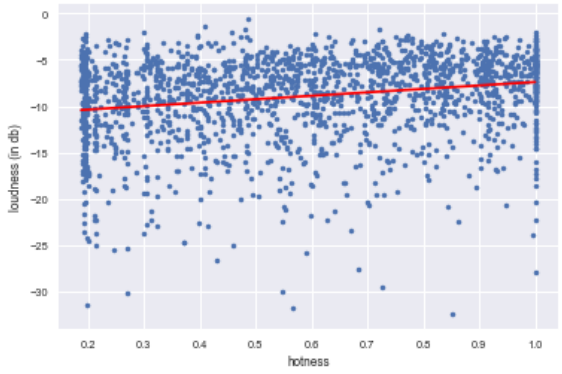

Figure 4: Loudness/hotness correlation scatter plot

4.2 Lyrics

The lyrics are an integral part of a song. They allow an easy recognizability together with a melody. They are also one of the features most easily reproduced by people when singing. One disadvantage is that not all music comes with lyrics, we will hereby only be focusing on *songs* in the literal meaning.

The musiXmatch dataset contains only about 237'000 songs, compared to the whole dataset which is about 1 million. Therefore if we want to consider lyrics as a feature for prediction, we will be heavily reducing our data set, unless we want to introduce some null-value into our model. The dataset is based on a bucket of words of the 5000 most prevalent stemmed words in the lyrics from the songs, therefore we are unable to do a semantic analysis. By (Logan et al., 2004) we can suspect that the analysis of the music is more effective than that of the semantics.

Our method to vectorize the lyrics is to recover the top 500 most represented words (filtering English stop words) from the occurrence totals and construct a vector from the occurrences of those words in each song.

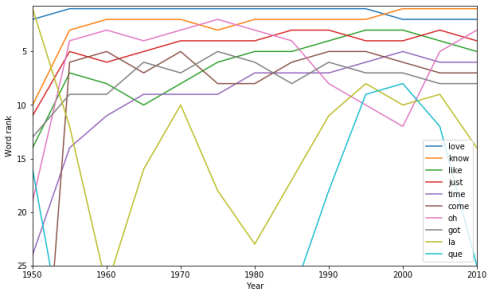

Figure 5: Rank of the globally most represented words over time

In addition to the ML vectorization, we also made an analysis of the usage of the globally most common words over time (Figure 5). We can see that most words are stable. We can discard the values before 1960, as there may be not enough data present to represent it correctly. Also we see a big increase in the usage of the word "que/qu", this can be explained by the lack of representation of the data: 1 million songs does not contain each and every song ever made. So old, foreign songs may not be included.

We also generated a word cloud based on the occurrences of the words globally, as seen in Figure 6.

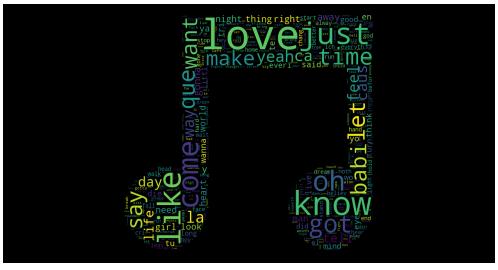

Figure 6: Word cloud of most occurring lyrics

4.3 Tags

The artist_terms and artist_weight in the dataset gives an array of tags specific to each song, with their associated weight. Amongst all the data, they are the ones that allows us to get a grasp of the song type. We will use these tags to find if there is a relation between the song genre (electro,rock ...) and the hotness in our ML model.

As the tags are strings, with a total of 7643 unique values we reduce by adding the weight for each tags, and then pick the 50 most frequent amongst our dataset. We then create a vector of 50 booleans for each song (if the tag is present or not) to analyze the relation between the tags and the hotness.

4.4 External summer hit ranking

The aspect of summer hit is not defined in our dataset, as we only have the release year. To complete the data, we scrapped a human made ranking giving the top 10 summer songs between 1958 and 2011. We then matched the titles with our dataset, with a performance of 485 over 540 songs.

4.5 Machine learning

Putting it all together, we combined the features of the dataset, the lyrics and the tags into a machine learning problem. For each song we associate the features list: duration, key, loudness, tempo, time signature, lyrics vector, tag vector. We applied a simple linear regression, we were motivated by the results of the correlations with the loudness, and hopped that the linear combination could give some results. The motivation also to use linear regression, is to get a results that would be interpretable. In the output weighted vector, we can consider values above 0 to be beneficial to the song hotness, and the negative values to be the features to avoid. Any algorithm used here should give more insight on the pattern between song hotness and song features.

To try and get the best results, we also used stratified sampling in the same manner as in the scatter plot in the correlation part. The idea motivating this choice is to avoid having the machine learning algorithm to be dragged towards the overly represented songs. In this case, we would get more songs with hotness around 0.5 then not hot songs (songs bellow 0.2), and hot songs (songs above 0.2). We took 10 classes, with approximately 1000 songs per class.

The value to beat is 0.25 RMSE (Root Mean Square Error). Looking at the histogram, an algorithm that would always give a hotness of 0.5 would have an RMSE of 0.25.

5 Results

We ended with 0.257 RMSE, so we are at the level of the minimum value to beat. We can interpret the final output weighted vector as the characterization of a 0.5 hot song. The negative values gives us features to avoid, and the positive the ones to promote.

Following this idea, tempo and duration should be promoted, and loudness should be decreased. For the lyrics, english words tend to have positive values, while words in other languages have negative values. The words with the most positive weights are 'know', 'oh', 'like', 'just', 'love'. Having 'love' and 'oh' with positive values in some way makes sense, many love songs talk about love, and just using 'oh' in a song is very common. In the other words worth noting in the words with the highest weights: 'babi' (which is probably baby that has been badly stemmed), 'feel', 'girl', 'heart'. The tags that have the most positive values are: pop, rock, indie, acoustic, alternative, indie rock, punk, soundtrack, metal. The one with the most negative values are: 'synthpop', 'germany', 'house', 'latin', 'world', 'trance', 'disco', 'soft rock'.

The features we extracted and the linear regression where not enough. Either the features where not well chosen, some better features expansion and preprocessing would of given better results, or linear regression is not good enough to model hotness with features.

6 Conclusion

One main reason for this result is that the Million song dataset was constructed in the goal to have the most diverse features possible. From this point, we should have expected that a model on the whole dataset would not give good results. A better approach would have been to create subsets, by year or by musical genre, to see other correlations with popularity (trends for example). An attempt to make a subset of summer hits, according to the billboard ranking

References

[Bertin-Mahieux et al.2011] Thierry Bertin-Mahieux and Daniel P.W. Ellis and Brian Whitman and Paul Lamere. 2011. *The Million Song Dataset*, ISMIR 2011.

[Herremans et al.2014] Herremans, Dorien and Martens, David and Srensen, Kenneth. 2014. *Dance Hit Song Prediction*.

[Logan et al.2004] B. Logan and A. Kositsky and P. Moreno. 2004. *Semantic analysis of song lyrics*, ICME 2004. DOI: 10.1109/ICME.2004.1394328

[McVicar et al.2011] McVicar, Matt, Tim Freeman, and Tijl De Bie. 2011. *Mining the Correlation Between Lyrical and Audio Features and the Emergence of Mood.*, ISMIR 2011.