

---

# Exploring the Development of Music over Time

---

Leander Zimmermann

Matrikelnummer 4165446

leander.zimmermann@student.uni-tuebingen.de

## Abstract

As part of their API, Spotify offers *audio features* for all their songs. In this report we use a dataset containing said audio features of just over 1.2 million songs to *predict the release year* from them with regression models. We then look at the resulting coefficients and try to draw conclusions, *how these features developed over time*. The corresponding repository can be found on <https://github.com/Zleander/MusicalDevelopment>.

## 1 Introduction

Music is a very complicated concept and it is hard to grasp, what makes a person like certain arrangements of sounds and dislike others. Nevertheless it is undeniable, that there is a general trend, which kinds of music tend to be popular. Interestingly this trend seems to change over time, with new genres developing and old styles of music diminishing in popularity.

This report will try to make a connection between basic features of songs and the year they the song was released. To achieve that we will build regression models to predict the release year from said features and then look at the coefficients to get a sense of which features were prominent in which periods of time.

## 2 Background

### 2.1 Audio Features

The features of the songs analysed in the report are generated by Spotify. A detailed explanation of every feature is available on the Spotify API reference [spo]. Here are all of the features we used:

*explicit, danceability, energy, loudness, mode, speechiness,  
acousticness, instrumentalness, liveness, valence, tempo, duration*

Most of them are somewhat arbitrary measures of things that can pretty easily be described by a human when listening to the music, but are very hard to put into numbers. Examples of this are *valence* (i.e. how happy the song sounds) and *danceability*. Others are concrete measurements, like the *loudness* and the *duration*.

### 2.2 Linear Regression

Linear regression is a way to fit a function onto some given data. In our case the inputs of the function are the features mentioned in section 2.1 and the desired output is the release year of the song. The simplicity of this formula can be seen in (1), with the feature vector  $x$ , the weights  $w$ , and the target  $t$ .

$$t = w_0 + x_1 w_1 + x_2 w_2 + \dots + x_n w_n \quad (1)$$

$$t = w_0 + x_1 w_1 + x_1^2 w_2 + x_1 x_2 w_3 + \dots + x_n^2 w_m \quad (2)$$

The weights are then found by minimizing the  $L_2$ -loss. When the relationship between features and target is more complicated than linear, one can augment this function by introducing a more complicated feature vector. The regression function with a polynomial feature vector of degree 2 is shown in (2).

### 2.3 Logistic Regression

Logistic Regression is very similar to Linear Regression. However it is a bit more complicated. The function we seek here, is no longer supposed to output a single number, but rather the probability of some amount of discrete events. From there the prediction can then be made by choosing the event with the highest probability.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (3)$$

$$p = \sigma(w_0 + x_1 w_1 + x_2 w_2 + \dots + x_n w_n) \quad (4)$$

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (5)$$

When the problem is binary, i.e. there are only two classes, a sigmoid function (3) is used to arrive at the probabilities, as in (4) with probability  $p$ . For more than two classes, there are two possibilities. Either each class is considered by itself in a binary problem, and the probabilities are normed in the end, or the probabilities are instead calculated using a softmax function (5). In the second case, which will be used here, the weights are found by minimizing the cross entropy loss.

## 3 Methodology

All experiments were conducted using pandas [Wes McKinney, 2010] and scikit-learn [Pedregosa et al., 2011].

### 3.1 Dataset

The dataset, created by Figueroa [2020] is a collection of over 1.2 million data points, each representing a song from spotify. For each song it includes the title, album, artist and the corresponding ids. More importantly, it also contains a release year ranging from 1900 to 2020 and the spotify audio features explained in section 2.1. Because less than 5% of the data points have a release year earlier than 1990 we decided to only use the more densely populated years 1990-2020. The data was then centered to 0 and scaled to unit variance to ensure comparable coefficients, and divided into a train- and a test-set in an 80-20 split for validation.

### 3.2 Models

To predict the year from the selected features, four different models were tested. We built a linear regression model (see section 2.2) which outputs a floating point number between 1990 and 2020 and a logistic regression model (see section 2.3) which treats the years as discrete categories. Because the relationship between the features and the release year may be way more complicated than a linear function could capture, both approaches were also tested on a polynomial feature vector of degree 3 (this results in 454 combinations of the original 12 features).

The evaluation of the models is achieved by calculating the the mean absolute distance of the prediction to the ground truth value. While the logistic regression model does directly minimize the distance to the ground truth, a better performance of the model will still show in a lower average distance. Additionally, the accuracy is calculated for the logistic regression models. Because the linear regression model do not have discrete outputs, calculating their accuracy would not make sense.

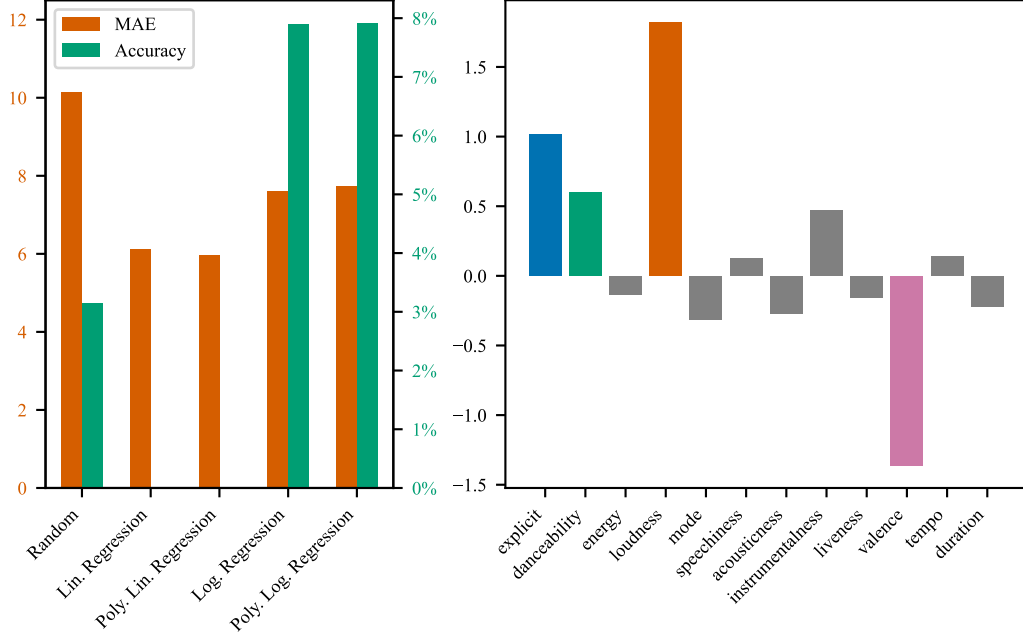


Figure 1: *Unfair* comparison of the mean absolute error (MAE) in years (lower is better) and accuracies (higher is better) of the Predictors (left) and the coefficients of the linear regression model (right).

## 4 Results

### 4.1 Models

The left side of Figure 1 shows some evaluation metrics of the models, performed on the test set. From the comparison to randomly generated numbers we can see, that none of them are particularly good. Considering the complicated nature of the data, however, this is not very surprising, as the model has to deal with countless outliers and the fact that it only has very few very basic features at hand. Another thing that is clearly visible, is that the models using the polynomial feature vector barely outperform their linear counterparts.

### 4.2 Coefficients

Because it is much more complicated, while not providing significantly better performance, we will ignore the polynomial feature vector for the purpose of analyzing the coefficients. Looking at the coefficients of the linear regression model can already tell us some things about the development of the corresponding features over time. Figure 1 shows these coefficients. For example we can now say, that a higher value of the feature "*loudness*" results in a higher output of the model. Thus, in reverse this means that the model assumes songs that have been released later to have a higher "*loudness*"-value.

The logistic regression model gives us a coefficient for each feature in every year, as the output of the model is now a vector of probabilities for the song being released in each year. These coefficients can be seen in the top part of Figure 2. While the relationship is not linear, a high coefficient still means a positive correlation between feature value and resulting probability. By looking at this plot, we could for example interpret that, according to the model, in the early 90s many songs have a high valence (i.e. sound happier), while recently more songs with a low valence were released. Most of the coefficients are very close to 0 in all years and only a few, which can be assumed to have high predictive value, have high and low values in different years. In contrast to that, the mean value per year of the features, displayed in the bottom of Figure 2, does not show the same features with high

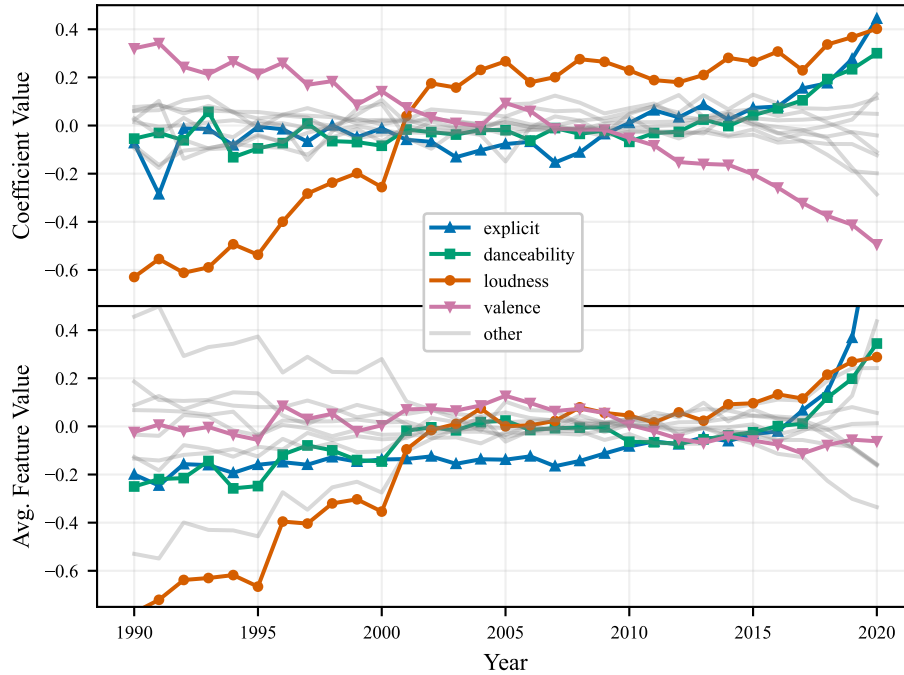


Figure 2: The coefficient for each year produced by the logistic regression model (top) and the average value of the features in each year (bottom). The colored lines represent the coefficients with the highest values in many years and their corresponding features. This makes them more important for the logistic regression predictor than the remaining features (grey).

predictive values standing out at all. Additionally, some curves, like the *valence* coefficient show trends, that the mean does not.

## 5 Conclusion

From the resulting coefficients, one could make the conclusion that in the 90s many songs were relatively happy and quiet, while in the late 2010s the music is much sadder, louder with more songs being rated explicit. While this may or may not be true, the experiment conducted here does not support assumptions like that. Between the arbitrary features the dataset that we know very little about, and the predictions of the models not being very good. We can not confidently conclude anything about the development of music.

## References

- Spotify audio features API reference. <https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features>. Accessed: 01.02.2022.
- R. Figueroa. Spotify 1.2m+ songs. <https://www.kaggle.com/rodolfofigueroa/spotify-12m-songs>, 2020. Accessed: 21.01.2022.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2011.
- Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a.