Spotify: analysis of song popularity, with reference to musical attributes and platform presence

Antonio Ciociola, Emanuele Rossi, Antonio Andrea Salvalaggio

May 20, 2024

Abstract

Song popularity can be affected by many variables, only a small portion of which we can reliably quantify. In this short work we try to evaluate what some of the most readily available quantitative data can tell us about its popularity.

Contents

1 Introduction

A song's popularity is tied to many variables, most of which can be hard to quantify. The most readily available data about a song is its spread. As today the vast majority of people listen to music through various streaming platforms we can use the data they publish as a good approximation for the whole population. Information about a song's position on a given platform's charts and about the number of official playlists that include it can be used to evaluate the spread of a given song. At the same time some of the musical characteristics of a song can be extracted from its digital representation using automated tools.

We aim at estimating a song's popularity, represented by the total number of times it's been listened, from its spread and musical characteristics. We expect to find high correlation between a song's presence on various platforms, i.e. its spread, and its popularity. We'll also evaluate the influence of various song's attributes. Since a song's absolute popularity is also tied to its genre, we expect at least some of them to be somewhat relevant.

2 Methods

Both the preprocessing work on the dataset and the numerical application of statistical methods have been done using the software *R* [R C23]. More information about the statistical methods used can be found at [FS24].

2.1 Preparing the dataset

We used the dataset "Most Streamed Spotify Songs 2023" [Elg23] which contains data about the 952 songs most streamed on spotify in 2023. For each song we have:

- General information like name, artist(s) and release date;
- Spread information in the form of chart position and number of official playlist containing the song on Spotify, Apple Music, Deezer and Shazam;
- Musical attributes automatically calculated by Spotify: *bpm*, *key*, *mode*, *danceability_%*, *valence_%*, *energy_%*, *acousticness_%*, *instrumentalness_%*, *liveness_%*, *speechiness_%*.

The dataset required some preprocessing in order to be used with the methods described below. We had to either remove text based features or convert them to numerical values:

- *track_name* and *artist(s)_name* could not be converted and were therefore removed;
- *key* was converted to the corresponding frequency in Hz;
- *mode* became either 0 (minor) or 1 (major).

As the data on key and shazam charts was partially unavailable we had to either remove the features or fill in the gap. As key's variance was extremely low and its contribution negligible and the shazam charts position was highly correlated to the other chart's positions we decided to carry on the analysis without these two features.

We also divided the dataset into a training and a test set, comprised of 85% and 15% of the samples respectively. Since we used cross validation to find the best value for the parameters of the regression, an explicitly separate validation set was unnecessary.

2.2 PCA

Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of a dataset while preserving as much variability as possible. It works by identifying the directions, called principal components, along which the data varies the most. These components are orthogonal to each other and are ordered by the amount of variance they capture from the data. PCA transforms the original data into a new coordinate system, where the first principal component has the highest variance, followed by the second, and so on. We are going to use it to get insights on the correlation between features and to check if there are redundant ones.

2.3 Ridge & Lasso

We would like to remind you that our ultimate goal is to estimate the β coefficients through which we can estimate our target variable y according to the following formula:

$$\vec{\beta}^T \cdot \vec{X} = \vec{y}_{ext} \tag{1}$$

To achieve this, the simplest method is the well known LS (Least Squares), where we estimate the p coefficients of the features by training the model on the dataset so that the quantity given by

$$\left\| X \cdot \vec{\beta} - \vec{y} \right\|_2 \tag{2}$$

is minimized. Ridge and Lasso, as we well know, also aim to minimize a function that resembles the sum of squared residuals, but to this function they add a penalty. In the case of Ridge, it consists of adding the L2 norm of the beta coefficients weighted by a factor λ, while in the case of Lasso, the penalty factor will be the L1 norm also multiplied by a coefficient λ.

$$\beta_{Ridge} = argmin(||\vec{y} - X \cdot \vec{\beta}||_2 + \lambda \cdot ||\vec{\beta}||_2) \tag{3}$$

$$\beta_{Lasso} = argmin(||\vec{y} - X \cdot \vec{\beta}||_1 + \lambda \cdot ||\vec{\beta}||_1) \tag{4}$$

Under what circumstances do we decide to use Ridge and Lasso?

Ridge is mainly used when the features under analysis are highly correlated with each other with the goal of reducing the variance on $\vec{\beta}$. Lasso is adopted when the number of features is extremely high compared to the number of available samples. Essentially, this method performs implicit feature selection by setting to zero all the coefficients of the features that are redundant.

Compared to LS, the main effects of these two approaches are essentially twofold: on one hand, the variance of the beta coefficients is reduced and overfitting is avoided more effectively, while on the other hand, a bias is introduced compared to LS.

Regarding our dataset in more detail, we decided to apply Ridge because, in our opinion, a significant fraction of the features are correlated with each other (see the correlation matrix (3.1)). We also decided to apply Lasso to understand which features can be considered negligible compared to the others.

To determine the λ to use in order to minimize the mean square error we applied 10-fold cross validation. All calculations have been done using the glmnet package ([FTH10] and [TNH23]).

3 Results

3.1 Correlation Matrix

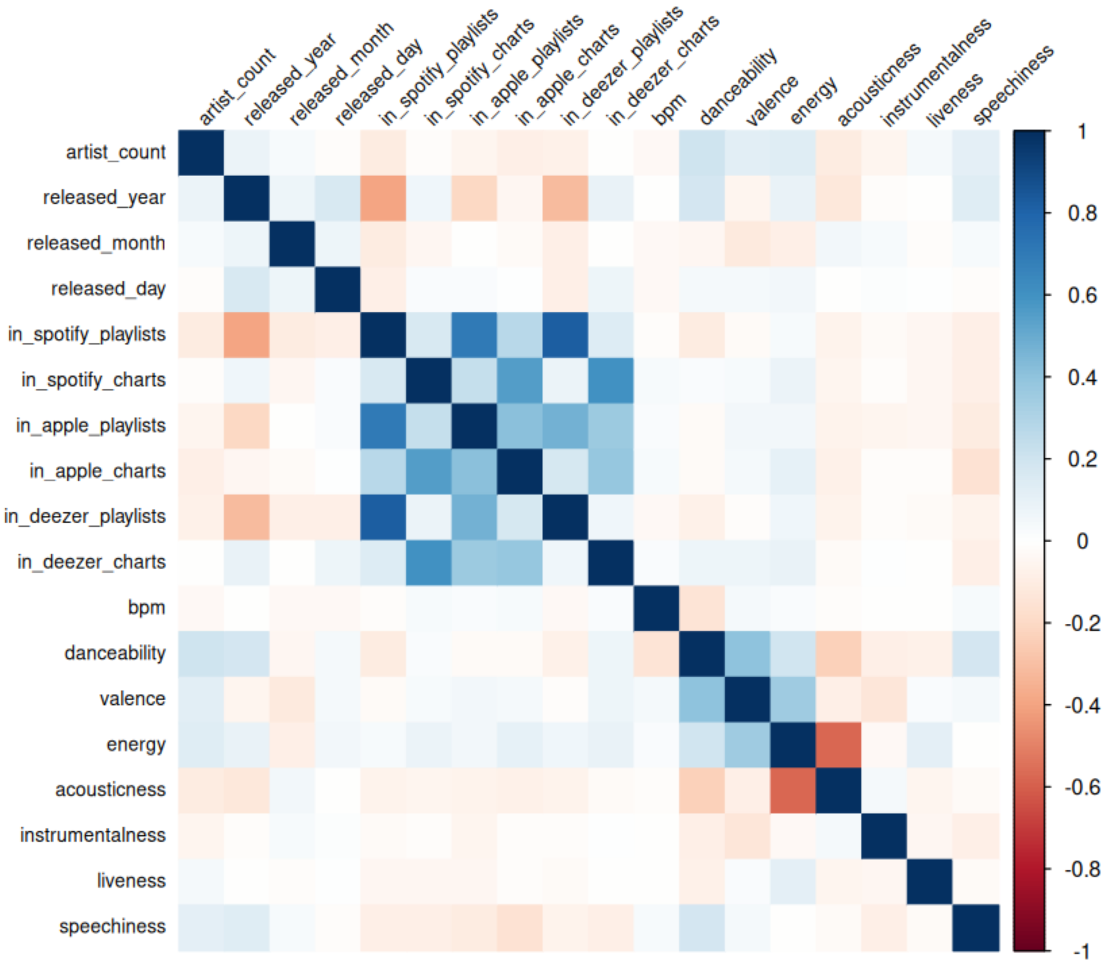

Figure 1: The correlation matrix of the dataset after preprocessing

In Figure 1 we can see the correlation matrix of our dataset (obtained using [WS21]). It is important to note how most of the charts and playlist features are highly correlated across various platforms. We'll account for this using Ridge regularization in the regression. Predictably, also some of the musical attributes, like acousticness and energy, are correlated with each other.

3.2 PCA

After running PCA on the dataset we obtain the graphs in Figure 2 and 3. We can see that the pve seems to be about 5% for every dimension. The same thing can be seen in Figure 3 where we can't recognize an elbow. It is therefore not possible to do dimension reduction without significant loss.

3.3 Regression

Now we will take under analysis the results of the regressions with penalization performed.

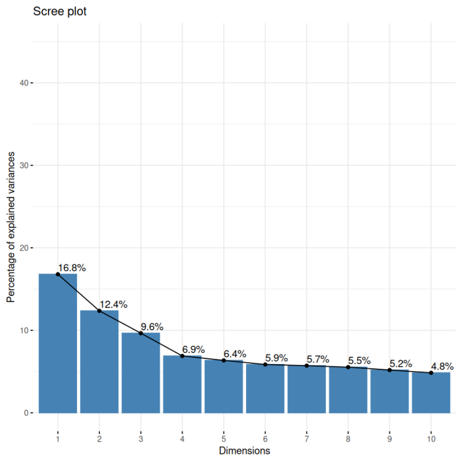

Figure 2: Graph with the pve of each dimension

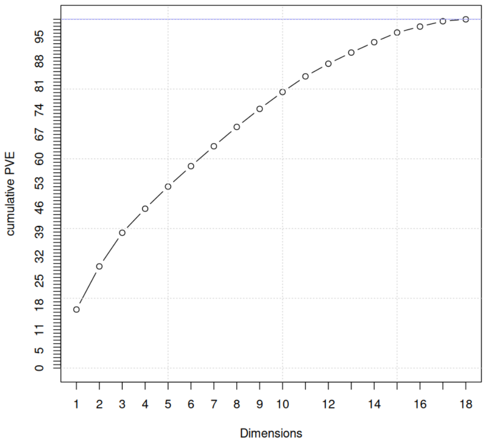

Figure 3: Graph with the cumulative pve

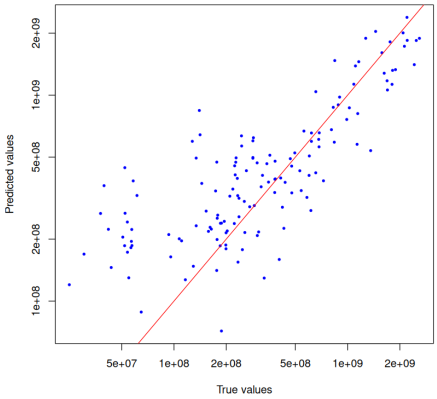

Figure 4: Prediction results after ridge regression on a linear scale

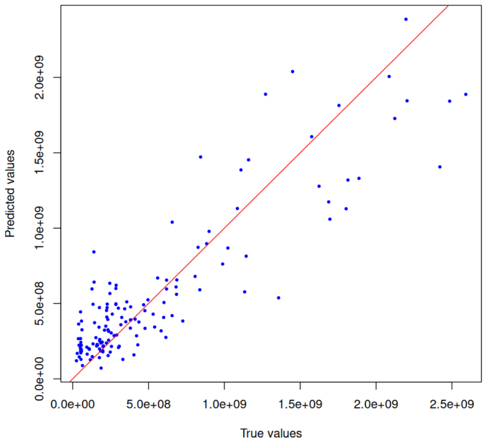

Figure 5: Prediction results after ridge regression on a log scale

3.3.1 Ridge

The figures 4 and 5 show the results obtained on the test set after finding the beta parameters in both linear and logarithmic scales: obviously, we would like the blue points to all lie as close to the red line as possible.

Figure 6 shows the tuning process of the regularization parameter λ. The procedure we adopted is as follows: we performed a sweep of the lambda parameter where, for a fixed value, we sought the vector $\vec{\beta}$ that minimizes the MSE. Having done this, we plotted the minimum MSE as a function of the logarithm of λ.

We note that the function has a minimum at $\log(\lambda) = 18$. For this value of the parameter, we found the corresponding vector $\vec{\beta}$ that minimizes the value and plotted it in Figure 8. From this graph, which has logarithmic scale on the y-axis, we can deduce that the number of streams is strongly positively correlated with the presence of the track in different playlists, which, of course, makes intuitive sense.

Finally, the last graph concerning Ridge shows us how the β coefficients shrink to zero while λ is increasing.

3.3.2 Lasso

The graph in Figure 10 shows the results obtained after performing regression with Lasso penalization. As we can see from figure 9, for $log(\lambda)$ approximately equal to 16 we have the minimum

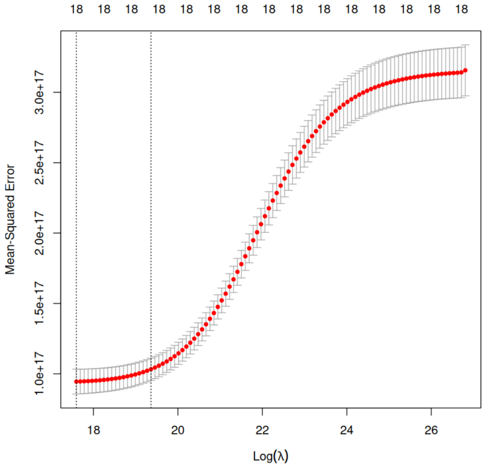

Figure 6: Cross Validation results for Ridge penalization

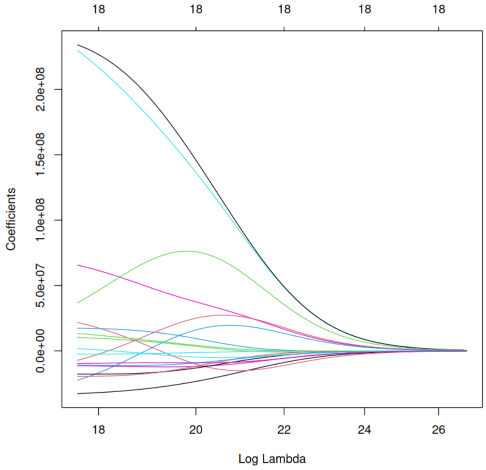

Figure 7: Plot of the coefficients in relationship with λ

value for the MSE. Figure 10 instead shows how the coefficients of $\vec{\beta}$ collapse to zero while λ is increasing. It's worth noting that the last coefficients to shrink to zero are the presence in apple and spotify charts, suggesting how these two features are the most relevant in determining our objective variable y.

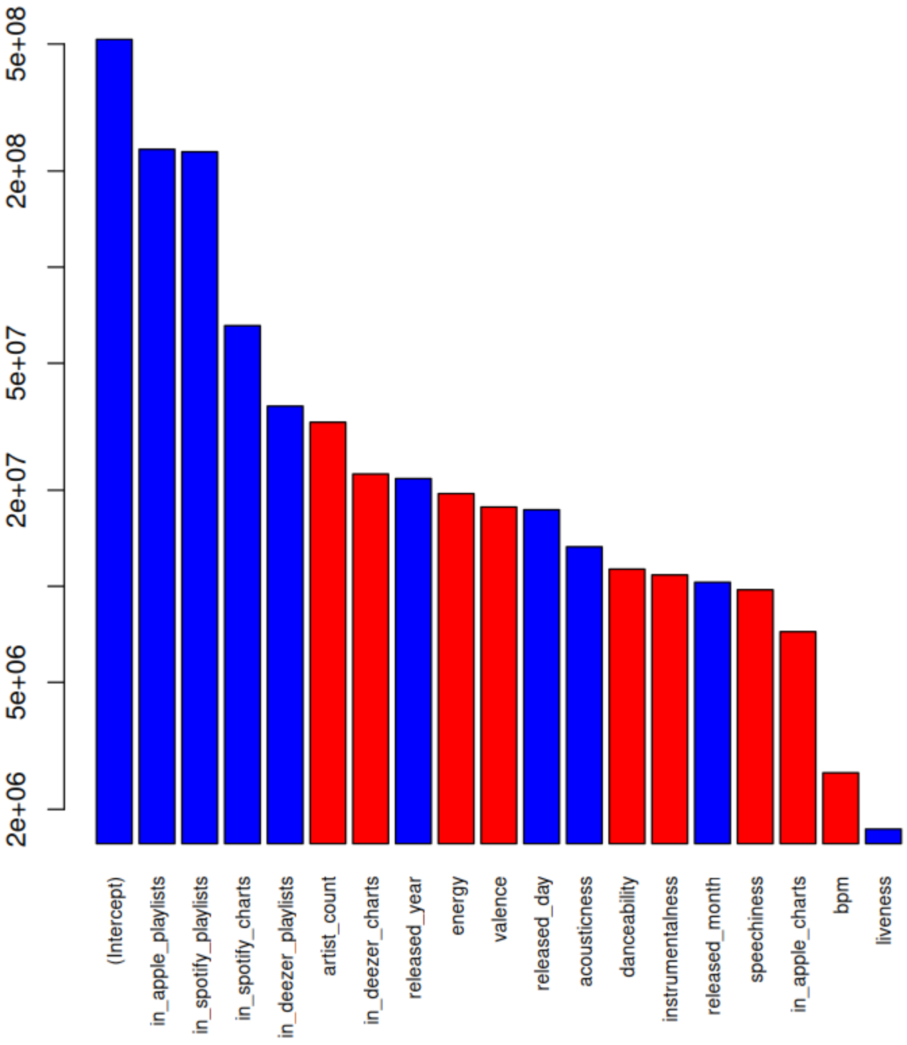

Figure 8: β parameters resulting from the regression with Ridge penalization. Red bars correspond to negative parameters

4 Discussion

4.1 Conclusions

As can be clearly seen from the $\vec{\beta}$ obtained with the regression (8), a song's spread across the various platforms is, unsurprisingly, the most important factor in determining its popularity. Unexpectedly, even tough we use the total number of streams to represent a song's popularity, the number of apple music playlists seems to be slighlty more telling, probably due to the more exclusive nature of the service.

Musical characteristics don't seem to be particularly relevant. The most important one seems to be energy: too energetic songs don't seem to be particularly appreciated.

Another notable contribution is a song's release year: unsurprisingly songs released more recently, especially this year, seem to be more popular than songs released in the past. Unexpectedly the number of artists present in a song also seems to negatively influence its popularity even though one would likely think that more artists usually mean a larger fanbase potentially getting to know it.

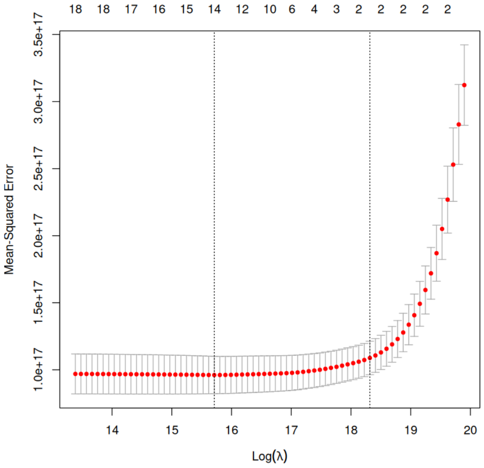

Figure 9: Cross Validation results for Lasso penalization

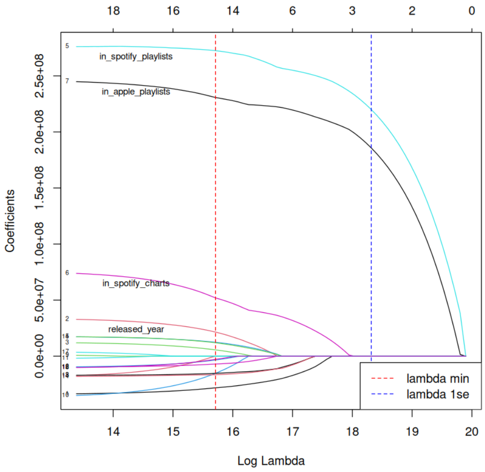

Figure 10: Collapse of the β parameters at the increase of λ

4.2 Future Work

The relationship between musical characteristics and popularity should probably be explored further, maybe taking into account a possible differentiation by genre. Another possibly interesting analysis could be to confront Ridge and Lasso penalization with the classic LS method in order to understand which algorithm performs best in terms of MSE.

References

[Elg23] Nidula Elgiriyewithana. Dataset: Most streamed spotify songs 2023. https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023, 2023. Accessed: 2024-03-14.

[FS24] Francesca Chiaromonte and Simone Tonini. Material for the course ”statistical learning & large data”. https://github.com/EMbeDS-education/ComputingDataAnalysisModeling20232024/wiki/SLLD, 2023/2024.

[FTH10] Jerome Friedman, Robert Tibshirani, and Trevor Hastie. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[R C23] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023.

[TNH23] J. Kenneth Tay, Balasubramanian Narasimhan, and Trevor Hastie. Elastic net regularization paths for all generalized linear models. *Journal of Statistical Software*, 106(1):1–31, 2023.

[WS21] Taiyun Wei and Viliam Simko. *R package ’corrplot’: Visualization of a Correlation Matrix*, 2021. (Version 0.92).