
Assessment of Spotify Chart Preferences

Florian Kriegel

Matrikelnummer 5746680

florian.kriegel@student.uni-tuebingen.de

Jonathan Währer

Matrikelnummer 5776535

jonathan.waehrer@student.uni-tuebingen.de

github: https://github.com/Fkrgl/Kriegl_Waehrer_spotify

Abstract

In this project we aim to assess feature importance with regards to the popularity of a song. For this we collected a data set of all available top 50 charts playlist on spotify for different countries and applied dimensionality reduction (t-SNE) in combination with different coloring scales based on certain feature values. Additionally, two regression models (linear and logistic) were trained with the goal to predict the popularity of a song based on the collected features and to classify whether a song of interest can be deemed as *chart material*.

1 Introduction

The music streaming platform Spotify is the most popular music streaming service nowadays. With over 381 million users, it provides a sound representation of today's most successful songs [1]. Thus, song data from Spotify can be used to analyse features of the currently most popular songs from all around the world. Spotify provides its own popularity measure to access song popularity based on play frequency and currentness [2].

In this report, we aim to identify important features for song popularity and use them to predict song popularity. For this, we acquired a data set of popular songs from the top 50 charts of all countries in Spotify and a Kaggle dataset with 1.2+ million songs (not part of the charts). Firstly, we used t-SNE to explore structure of the data. Next, linear and logistic regression models were trained to predict the popularity of a song based on pre-selected features.

2 Materials and Methods

2.1 Data Acquisition

In a first step, we downloaded the top 50 chart of 51 different countries from Spotify as representative popular songs. For this, we used the Spotify developer's API [3] to download 20 meta and audio features for each song from the 51 chart playlists (see script `download_spotify_data.py`) resulting in a total of 2549 songs. In order to also consider unpopular songs, we downloaded a data set with 1.2+ million songs from Kaggle [4] (see script `download_kaggle_data.py`).

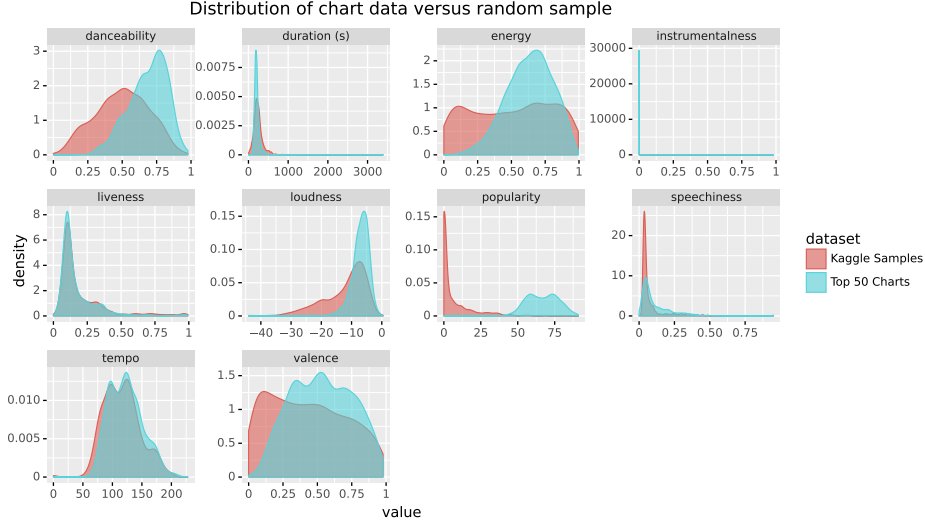


Figure 1: Distribution of selected features compared between chart and Kaggle data.

2.2 Data pre-processing

We extracted all audio features from our data sets and ended up with nine nominal and three categorical features for each track. Duplicated songs in the chart data set were removed and the nominal features were scaled using Z-score transformation resulting in 1060 unique chart songs.

We produced a balanced training set for regression and classification by randomly sampled 1000 songs from the Kaggle data set as examples for more unpopular songs. Finally, both data sets were merged and labeled according to their source (charts or Kaggle) resulting in a training data set with 2060 tracks.

2.3 Visual feature pattern analysis

Feature distributions were analysed using density histograms. To reveal more complex structure in the training data, we combined t-SNE with a color scaling on different features.

2.4 Regression and classification

A linear regression model was trained on two thirds of our data set. The performance of our model was accessed using the root mean squared error (RMSE) of a 5-fold cross validation on the training set and the test set. Further, a logistic regression model was trained to distinct between unpopular (popularity ≤ 50) and popular (popularity > 50) songs.

3 Results

3.1 Visual feature pattern analysis

Figure 1 shows that the chart data sets popularity distribution is shifted towards high popularity values while the popularity distribution of Kaggle songs is shifted towards lower values. The features duration, *instrumentalness*, *liveness*, *speechiness* and *tempo* have a fairly similar distribution in both samples. Only the features *danceability*, *energy*, *loudness* and *valence* ('mood' of the song measured from zero for sad to one for happy [3]) also show noticeable differences in their distribution. Based on this observation, we decided to train our models in two batches: one for all features and one only for the four that differ most between the two groups (danceability, energy, loudness and valence).

We analysed the four features using t-SNE and color coded each feature to highlight gradients in the produced map. Figure 2a shows the t-SNE map on our training data color coded by the labels. Although, no distinct clusters become apparent, the color code reveals that chart tracks are over-

Table 1: Performance on test set for regression (RMSE) and classification (Accuracy) models

| model | performance (12 features) | performance (4 features) |
|---------------------|---------------------------|--------------------------|
| linear regression | 26.79 | 27.52 |
| logistic regression | 0.78 | 0.75 |

represented in the lower right quadrant while the more unpopular kaggle tracks tend to cluster more in the top and left of the map. The color coding for the four most distinct features in figure 2b show clear gradients in the t-SNE map. Here, we can see similar trends of the gradients of the features danceability and valence which show high values in the bottom left and low values towards the upper right. Thus, more danceable songs seem to also be more happy and cheerful (high valence). Both, valence and danceability gradients roughly follow the popularity gradient indicating that popular songs tend to have higher values for both features compared to unpopular ones. Noticeably, a high energy measure can be found in both popular and unpopular songs. This can also be explained with the corresponding density histogram (figure 1) and the uniform distribution of this measure for the Kaggle data set.

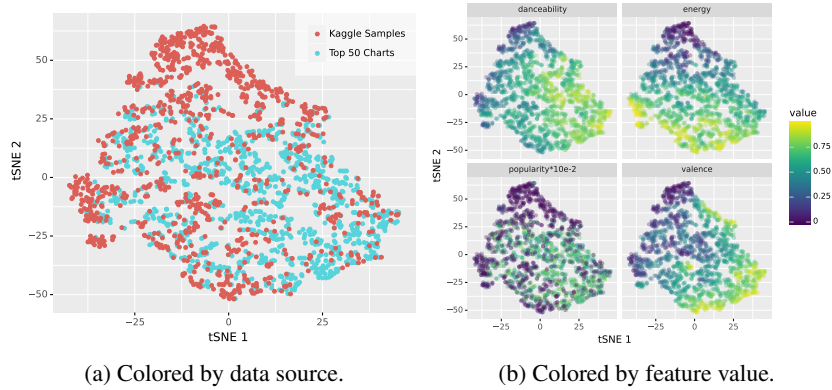


Figure 2: Application of t-SNE to our data set with color coding corresponding to data set (a) or feature value for the most important features (b).

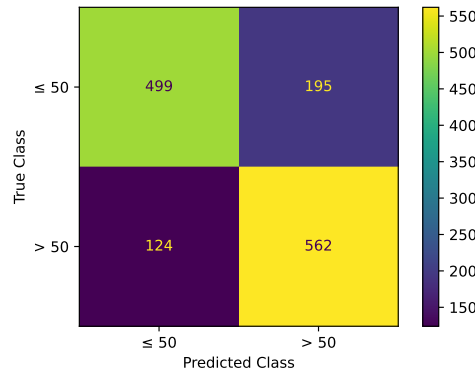
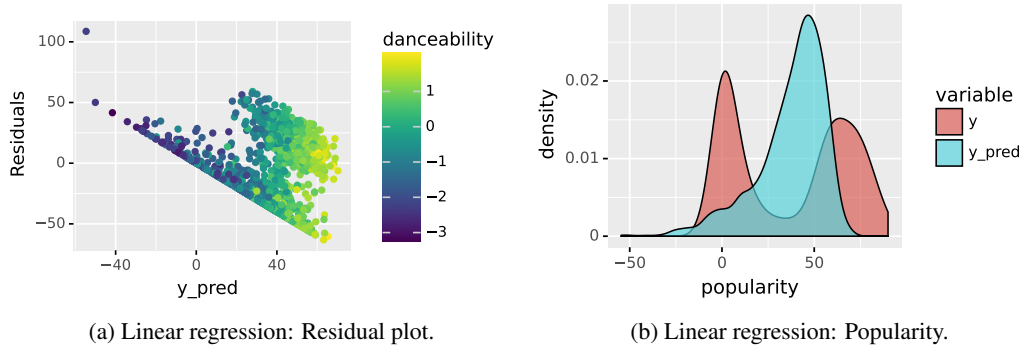
3.2 Popularity prediction

Figure 1 shows the performance of the linear and logistic regression models. For both models, the four most distinct features between the two groups achieved roughly the same performance compared to the models that used all 12 features. The residual plot for the linear regression model on four features in figure 3a reveals a clear pattern in the residuals of the training data. Further, the color coding shows that our model seems to regard danceability as the most important indicator for song popularity because it tends to over predict songs with high danceability and underpredicts songs with low danceability. By taking the very different distributions between true and predicted popularity in figure 3b into account, we conclude that the linear regression model is not capable of capturing the relationship between the features and the response.

According to the confusion matrix in figure 3c, the logistic regression performs fairly well in distinguishing between popular and unpopular songs.

3.3 Discussion

In this report, we aimed to analyse feature importance for the popularity measure of Spotify songs and used the results to train regression models to forecast song popularity. Although, the visual feature pattern analysis revealed some differences in feature distributions between popular and unpopular songs, no distinct clusters tend to form in the t-SNE projections. We suggest that Especially songs with low popularity score but a high number of clicks cause problems here.



(c) Logistic regression: Confusion matrix.

Figure 3: Results of the logistic regression

While we showed that logistic regression can be used in order to classify whether a song can be deemed as chart material, other models have been applied for similar tasks and might be worth looking into [5]. It might also be possible to further improve our logistic regression model by extracting and analysing more features from the Spotify API or by adjusting the cutoff for the class assignment. An improvement could also be achieved by adjusting the popularity-cutoff: Based on the distribution of the popularity (figure 1), the cutoff between the two data sets is slightly below 50 which might also improve our model. In conclusion, the aims of this project were partially fulfilled. While there were no distinct t-SNE clusters, color-coding helped to somewhat assess the relationship between popularity and other features to a certain extent. What's more, while there is room for improvement on the statistical model, the results of this project have demonstrated the capability of simple models to predict the chart potential of a song with somewhat reasonable accuracy.

References

- [1] *Spotify active users*. URL: <https://www.statista.com/statistics/367739/spotify-global-mau/>.
- [2] *Spotify song popularity*. URL: <https://towardsdatascience.com/predicting-popularity-on-spotify-when-data-needs-culture-more-than-culture-needs-data-2ed3661f75f1#:~:text=According%20to%20Spotify%2C%20E2%80%9Cpopularity%20is,a%20lot%20in%20the%20past.%E2%80%9D>.
- [3] *Spotify developers*. URL: <https://developer.spotify.com/>.
- [4] *Kaggle, Spotify data set*. URL: <https://www.kaggle.com/rodolfofigueroa/spotify-12m-songs>.
- [5] Carlos Vicente Soares Araujo. "A Model for Predicting Music Popularity on Spotify". In: *Recall* 50 (2020), pp. 173–90.