How to predict upcoming hits? Correlation between Spotify's popularity rating and its other audio features

Ivan Rätsch,* Noah Scheld,† Christian Willner,‡ and Mudassar Zahid§
Universität Hamburg, MIN-Fakultät, Fachbereich Informatik, Vogt-Kölln-Straße 30, 22527 Hamburg
(Dated: January 30, 2021)

While Spotify offers more than 144 Million songs[1], a sample with 160.000 entries might help to predict the popularity of a new track before it is even released. Tools from classic data–analysis are used to identify key characteristics which become a focus with machine–learning techniques to build a prediction engine.

I. INTRODUCTION‡

For record producers, musicians, writers and music labels it is of high interest to judge a song as early as possible. Which track should be published as a single, what song has the highest chances for success?

Spotify is labeling audio features for classification purposes. Success is not based on these characteristics. Marketing, band popularity and probably chance are huge factors to be considered as well. For this study these additional aspects cannot be considered, but the resulting metric might provide insight for planning marketing campaigns and other actions.

A. Research Goal

After hearing a song it might be very obvious to know if one likes it or not. Music professionals rely on their feeling to estimate the success of a given track and companies might have established metrics to gauge future interest in a track. While a lot of music in the 00's had to include rap parts to widen the appeal, producers might now push for Spanish lyrics to address a broader audience[7]. These decisions cannot be addressed with this study, but the characteristics of a song might help to give pointers for the creation of a popular song.

To research the connection between popularity and the aforementioned properties of a Spotify song two strategies are being evaluated. The first part in section II is a statistical evaluation to check for influences within the characteristics. As a second part a model for a prediction is presented in section V.

B. Data Corpus

The complete database of Spotify covers many genres, artists and songs from many different decades. The data set[2], that is used in this report represents this corpus pretty well. The data set covers:

- Characteristics of more than 170.000 songs,
- Entries from 1921 to 2021,
- 2.972 genres,
- 36.195 artists and
- 400–8.000 songs per year.

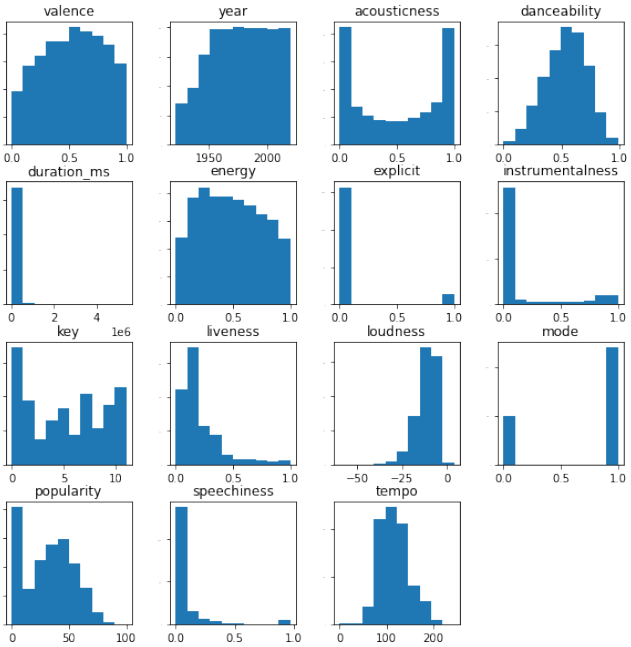

Figure 1. Data distribution.

Spotify itself has a recommendation algorithm based on song characteristics. The inner workings are not publicly known, but predictions are based on 14 key characteristics: valence, acousticness, danceability, duration, energy, explicit, instrumentalness, key, liveness, loudness, mode, popularity, speechiness and tempo. Mode and explicit are modal values of 0 or 1 and key is categorical. Duration is given in *ms* and popularity ranges from 0 to

* ivan.raetsch@studium.uni-hamburg.de
† noah.scheld@studium.uni-hamburg.de
‡ christian.willner@studium.uni-hamburg.de
§ mudassar.zahid@studium.uni-hamburg.de

100. Loudness is given in dB and tempo in BPM. The other audio features range from 0 to 1. An overview of the complete set can be seen in Fig. 1.

II. STATISTICAL EVALUATION[‡]

What song characteristics have an influence on its popularity?

Just as some people prefer certain genres of music, it makes sense to segment our data into genres in order to get a more precise picture. What might be considered good in one genre, might harm the popularity in another. An exemplary comparison between the genres *rap* and *rock* is shown in Fig. 2 and Fig. 3. The red polynomial regression line is of the third order to avoid overfitting. The confidence is shown with a wider light red area around the line. All graphs have their Pearson-correlation and p-values marked accordingly.

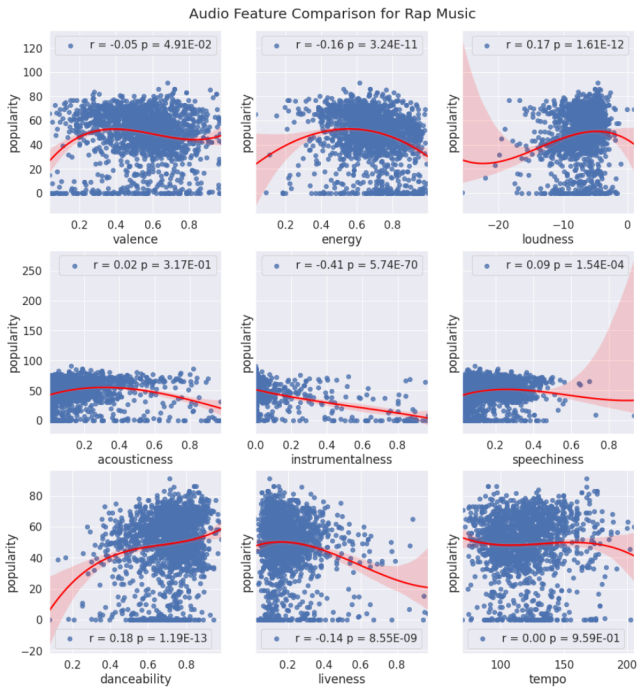

Figure 2. Effect on popularity for rap songs.

It can be argued that a certain level of valence is helpful for a popular rap song. Low danceability and high instrumentalness might be a hindrance though. Louder songs show a positive relation with popularity as well. Tempo and acousticness have a very minor influence in rap.

In comparison, rock tracks are not much influenced much by their tempo either and have a certain loudness level that should be avoided. Energy has a different maximum than in the result for rap songs. Perhaps surprising is the dislike for a high acousticness in rock music, which shows a negative relation to popularity of -0.36. Comparing this to the acousticness correlation factor in rap's of 0.02 shows the importance of a differentiation by genre.

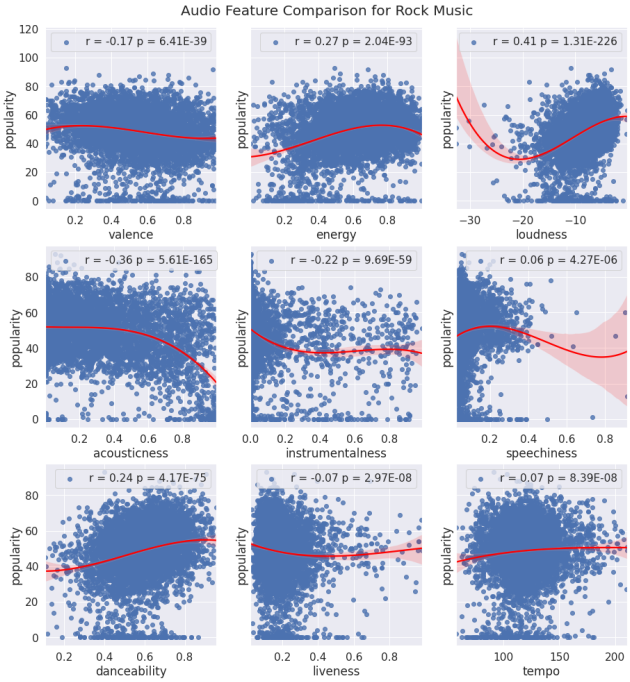

Figure 3. Effect on popularity for rock songs.

Many of these graphs suggest a strictly positve or negative influence from the audio artifact to the popularity. E.g. rap–musicians could adjust their songs for maximum loudness, minimum acousticness and high danceability. More interesting is a look at the polynomial regression curves with distinct maxima and Minima, like the energy level of rap–music in Fig. 4. The analysis suggest, that a rap–song should aim to achieve an energy rating of 0.675.

When one is looking at other genres in the energy to popularity comparison in Fig. 4, it is evident, that genres are indeed very different when it comes to their optimums. Another comparison is given for the influence on valence, the so-to-speak happiness, of a song in Fig. 5. While Soul songs constantly lose popularity with increasing valence, it is opposite for jazz pieces. Rap is having one of the highest absolute gains in popularity when increasing valence, while the difference from one extreme to the other in jazz is only moderate. In connection with the examples above, for the rap songs, the graph in Fig. 5 suggests an optimal valence at 0.39, while rock is preferred to be a little more on the sad side at 0.36.

These comparisons serve well for a first impressions, but they have to be handled with care. They can show a general trend, but the confidence is not shown. This can be seen for the speechiness curve in the rap–comparison in Fig. 2. Higher values have a very low confidence due

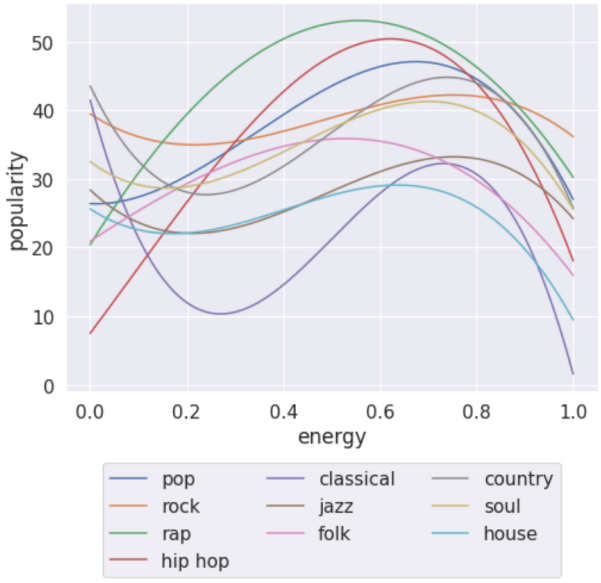

Figure 4. Influence of energy on popularity for the 10 most listed genres

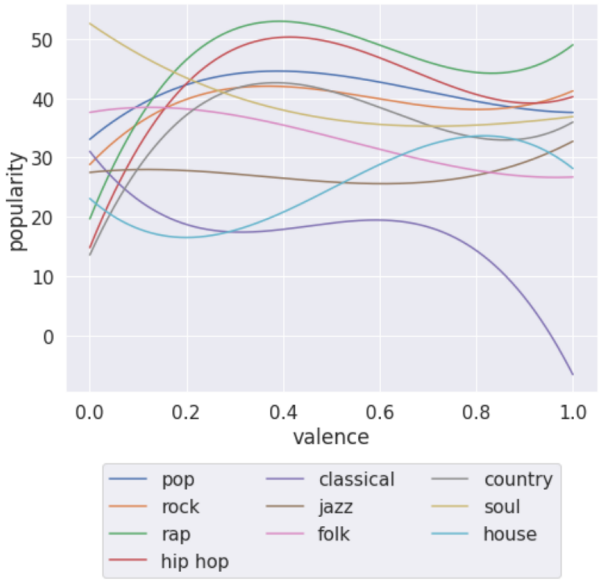

Figure 5. Influence of valence on popularity for the 10 most listed genres

the lack of data–points and for educated decisions both graph types should be used.
The feature comparison in Fig. 2 and Fig. 3 was realized with the seaborn[8] library, the polynomial fitting in Fig. 4 and Fig. 5 was performed with statsmodels[6] for python.

III. POPULARITY PREDICTION

Is it possible to predict the popularity of a song from its other characteristics?

While a statistical analysis with certain key aspects can be done with the method above, the influence of nine or more factors can be hard to model. This is a prime example of the usefulness of an AI network.

With supervised learning from the Spotify data the trained network should be able to guess a popularity for a given song. This could be fine tuned by giving a certain genre that the song is positioned in. If the genre is only sparsely represented in the dataset this option would lead to false results.

The ten biggest genres have more than 1000 entries each and might supply a decent base to train the network. A classification algorithm might help to solve that problem if a genre cannot be defined. On the broader scope a clustering strategy might decouple the AI completely from human classification needs, as certain genres might be closer for the algorithm than a human would have guessed.

IV. INTERIM CONCLUSION

From the evaluation of rock and rap songs above, it can be seen that popularity is affected by different key characteristics. This small insight seems to support the initial hypothesis from section II.

It is likely that a neural network could offer similar results and help to understand how the selection of a genre changes these key characteristics. In the end, one could produce the theoretical values for a perfect song in a given genre.

V. METHODS, APPROACH & IMPLEMENTATION*†§

In the smaller datasets we used, each song was assigned to a genre and some genres only had few songs. So, for the following methodical approaches we opted for a dataset with more songs but without a classification for genres.

In the following section we will explain the methods and approaches used to solve the regression problem with an accuracy of up to 83%. Each approach at solving the regression problem makes use of decision tree regression as well as random forest regression. In the end, we will compare the results of each approach followed by our attempt at building a neural network. The following subsections will outline the two types of regressions used in each approach.

1. Decision Tree Regression

A regression tree is a useful method to predict the target value when multiple predictors, i.e. our track attributes, are given. For each predictor, a binary regression tree is built where each value of the predictor is compared to its subsequent value. For each point in the data, the residual (difference between observed value and predicted value) is used as a metric of how accurate the predictions are. The root node for each predictor is chosen by plotting the residuals with the respective predictor value and picking the residual with the smallest value. After creating a regression tree for each predictor they are sorted, where the predictors with smaller residual values are given more weight than those with bigger residual values.

2. Random Forest Regression

A random forest regression is a special type of a decision tree. The model constructs multiple decision trees at training time and outputs the average value of those trees.

A. Regression Approach #1

For our first attempt at solving the regression problem, we used the original, unaltered dataset described in the introduction.[2] Our dataframe contained the scalable track attributes acousticness, danceability, duration in *ms*, energy, explicity, instrumentalness, key, loudness, liveness, mode, speechiness, tempo, valence and year while our target variable was popularity. First, we randomized our data and split it into an appropriate training and test size. Our first model was a decision tree regression. Using this method, we were able to create a model[3] with a mean absolute error of 8.706. With a random forest regression, the mean absolute error improved to 6.504.

The relative importance of our feature attributes in respect to the popularity attribute can be visualized using a *feature importances* plot on our random forest regression model.

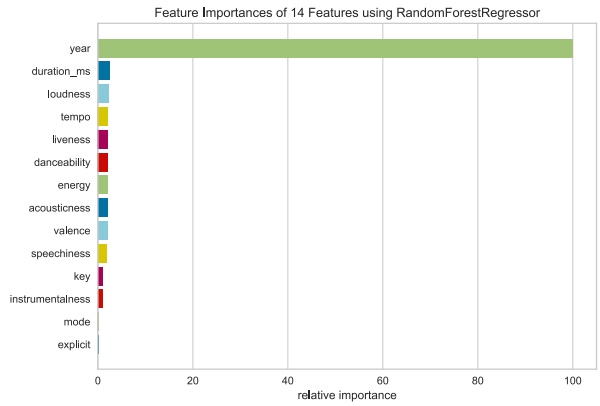

Figure 6. Relative importance of Feature Attributes

Surprisingly, *year* was the most important attribute by a wide margin. The reason for that becomes clear after looking at how Spotify defines our target value 'popularity'. It only accounts for popularity measured at the time of the dataset being created (2020). Naturally, songs that were released in 2020 are going to be more popular right now than songs released decades ago even if those songs were very popular at their respective time.

With that knowledge in the back of the head, it is not surprising that the random forest regression model guessed the right popularity with a high accuracy of 83%. This is because 'guessing' a low popularity for older tracks is a relatively easy task given how popularity is calculated. The following approach #2 tests this hypothesis.

B. Regression Approach #2

Looking for a way to test our hypothesis, whether year is the deciding factor due to how popularity is calculated, without artificially altering our dataset too much, we decided to remove tracks that were released prior to the year 2000 from our dataset. We anticipated this would lead to a mitigation of the effect the attribute *year* had on the accuracy of our predictions. After processing the data we were left with approximately 50,000 tracks. Creating a plot with the *feature importance* module on our processed dataset confirmed our hypothesis, as *year* was no longer the most important attribute.

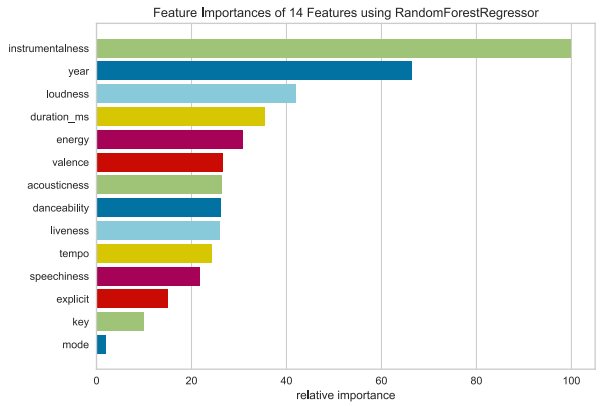

Figure 7. Relative importance of Feature Attributes

Using the decision tree regression method, we were able to create a model [4] with a mean absolute error of 14.980. the random forest regression model improved the mean absolute error to 12.714. As expected, the accuracy of our model went down to about 50%, as there are no longer, or rather far less, tracks whose popularity is easy to guess because of their release date. Although the accuracy went down, the model now works better for its intended purpose which is predicting popularity based on track attributes.

C. Regression Approach #3

Because of the relevance that *year* continued to have on the popularity, we altered the data so that we do not use *year* as a feature attribute anymore. Therefore, removing *year* should give us a clearer view of which attribute is overall more important in terms of popularity. The relevance of the other attributes changed as can be seen in the following graph:

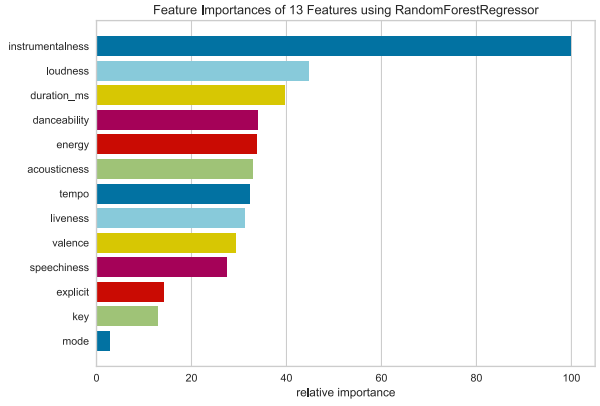

Figure 8. Relative importance of Feature Attributes

Using the decision tree regression method, we created a model[5] with a mean absolute error of 18.244. Using the random forest regression the mean absolute error was improved to 15.445. Even though the mean absolute error worsened compared to the previous approach, this model gives us helpful results. The year of release is not something that can be changed when an artist wants to write a new song, so this model gets rid of a that attribute.

D. Results of the three Regression Approaches

	MSE (DTR)	MSE (RFR)
Approach #1	8.706	6.504
Approach #2	14.980	12.714
Approach #3	18.244	15.445

Table I: Comparison of the Mean Absolute Error (*MSE*) for Decision Tree Regression (*DTR*) and Random Forest Regression (*RFR*) for the three approaches

A notable observation is that the random forest regression always delivered a better mean absolute error result than the decision tree regression in the respective approach. However, with each approach the model became less dependant on the attribute *year* which led to a decline of the *MSE* value. Although removing the dependency to the *year* attribute increases the variance of the model which may result in a better prediction with songs where the release date is unknown.

E. Neural Network Approach #1

For our first attempt at creating a neural network, we used the original, unaltered dataset. We configured our model using one flat layer and three dense layers with *relu* as activation function. Another dense layer with a soft-max activation function was used for the predictions layer. The result was a loss rate of 4.02 and an accuracy of about 16%.

F. Neural Network Approach #2

Our second attempt at neural networks used the altered dataset with tracks released between 2000 and 2020. Surprisingly, the loss rate and accuracy improved to 3.665 and 26.17% respectively. In contrast, the regression model approaches worsened with every modification made to the dataset.

G. Results of the two Neural Network Approaches

	loss	accuracy
Approach #1	4.02	16.04%
Approach #2	3.665	26.17%

Table II: Comparison of the loss rate and accuracy of both approaches

VI. CONCLUSION

The less importance we gave the feature attribute *year* the worse the mean absolute error became. This makes sense due to the fact that older songs have a worse popularity rating than newer songs, because the popularity was recorded in the year 2020. In the end, our models were able to guess the popularity with a mean error of around 6 to 18. That means that on average the regression models were about 6 to 18 points off the actual popularity rating. The neural network with a loss value of about 3 to 4 was even better at predicting popularity. In comparison to our regression approaches the neural network achieved a minimal loss even when removing old songs.

ACKNOWLEDGMENTS

Thank you stackoverflow!

[1] Spotify AB. *For the Record*, accessed 2020-12-10. https://newsroom.spotify.com/company-info/.

[2] Yamaç Eren Ay. *Spotify Dataset 1921-2020, 160k+ Tracks*, accessed 2020-11-22. https://www.kaggle.com/yamaerenay/spotify-dataset-19212020-160k-tracks.

[3] https://svgshare.com/i/TbZ.svg.

[4] https://svgshare.com/i/Tbu.svg.

[5] https://svgshare.com/i/Tbh.svg.

[6] Skipper Seabold and Josef Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[7] Ed Vulliamy. *She loves you... sí, oui, ja: how pop went multilingual*, accessed 2020-12-10. https://www.theguardian.com/music/2019/apr/06/latin-spanish-pop-takes-over-from-english-language.

[8] Michael Waskom, Olga Botvinnik, Drew O'Kane, Paul Hobson, Saulius Lukauskas, David C Gemperline, Tom Augspurger, Yaroslav Halchenko, John B. Cole, Jordi Warmenhoven, Julian de Ruiter, Cameron Pye, Stephan Hoyer, Jake Vanderplas, Santi Villalba, Gero Kunter, Eric Quintero, Pete Bachant, Marcel Martin, Kyle Meyer, Alistair Miles, Yoav Ram, Tal Yarkoni, Mike Lee Williams, Constantine Evans, Clark Fitzgerald, Brian, Chris Fonnesbeck, Antony Lee, and Adel Qalieh. mwaskom/seaborn: v0.8.1 (september 2017), September 2017.