Regression for Spotify Song Popularity

Midterm Project Report for DATA 1030, Fall 2021

https://github.com/Pooja-Barai/spotify-songs-regression.git

Pooja Barai

I. Introduction

Motivation: The music industry is exploding, despite and especially since the beginning of the Covid19 pandemic. The world has seen a rise in both the consumption and production of music, and understanding what factors are associated with popular music can be insightful for music producers and artists. Given that music is extremely subjective to a human ear, it would be interesting to see if machine learning models can identify useful patterns. Modelling in this way can also be generalised to other music business analytics, and can thus be useful for such future endeavours.

Dataset: This project attempts to use machine learning to conduct regression on the continuous target variable, popularity, in order to try to understand what factors lead to popular songs. This variable, a metric for the popularity of a soong which ranges from 0-100, is a derived metric calculated by the Spotify Web API itself. The dataset used for this project, however, is from Kaggle and not directly from Spotify. It contains audio features of 169,909 popular songs from the years 1921-2020, and each song has an associated id, name, artist, release date, year, and duration. The 12 audio features are acoustics, danceability, energy, instrumentalness, valence, tempo, liveness, loudness, speechiness, mode, explicit, and key. From these, 7 are derived metrics that range from 0-1, also calculated by Spotify itself. Together, these and popularity form the 19 features of this dataset.

Literature search: There are multiple authors have published projects that conduct exploratory analysis on this dataset, and who tried answering similar questions. In Ochi, Virginia et. al, the authors analysed the relation between danceability and popularity, using statistical regression models and time series analysis. They gained insight on questions like where danceable songs can be successful in the future, and whether these trends will continue. Another public project by Jesperhemmingsson, attempted to predict song popularity using Random Forest Regressor, SGD Regressor and XGBoost Regressor. They found that song popularity mostly depends on features like energy, loudness, danceability, and instrumentalness.

II. Exploratory Data Analysis

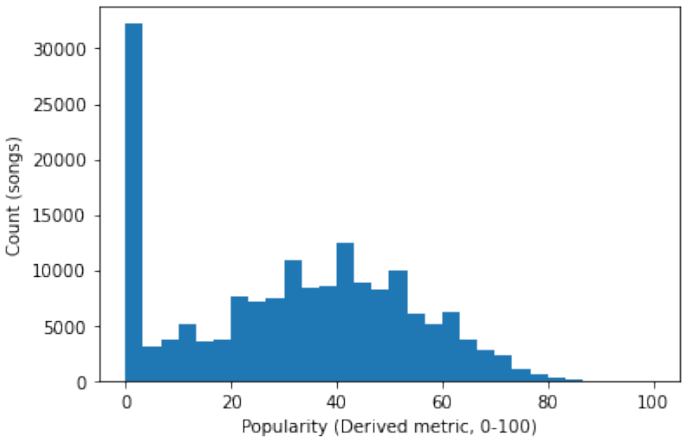

Fig.1. The histogram distribution of the count of songs for various popularity scores, ranging from 0-100.

During exploratory data analysis there were some plots that revealed interesting relationships, which have been included here. For example, Fig.1 highlights how a large number of songs have a popularity score of 0, with a tall peak at 0. Although the distribution appears to be bimodal, if we ignore the first peak at 0, the distribution appears to be mostly Gaussian, with a mean of approximately 40 and a large variance.

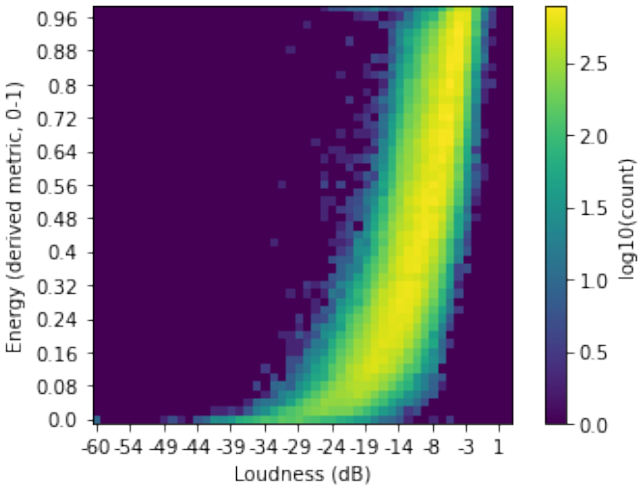

Fig.2. The heatmap distribution of the log10(count) of songs for a given energy score, which is a derived metric ranging from 0-1 and a given loudness, ranging from -60 to 0 decibels.

Fig. 2 shows that there is a strong positive correlation between the loudness of a song, and the energy score associated with the song. Intuitively, this makes sense given that a loud song would be associated with a song that feels more 'energetic' and it is interesting to see this experimentally visualised. The log relationship highlights the underlying nature of the unit of measurement for loudness, which is decibels.

Fig. 3 displays the interesting and surprising relationship between the popularity of a song, and whether or not it contains content that would be typically considered as explicit. Although one would expect that generally explicit songs are less popular, the distribution of explicit songs is shifted to the right, showing that the average popularity score of explicit songs is higher than the average popularity score of songs that do not contain explicit content.

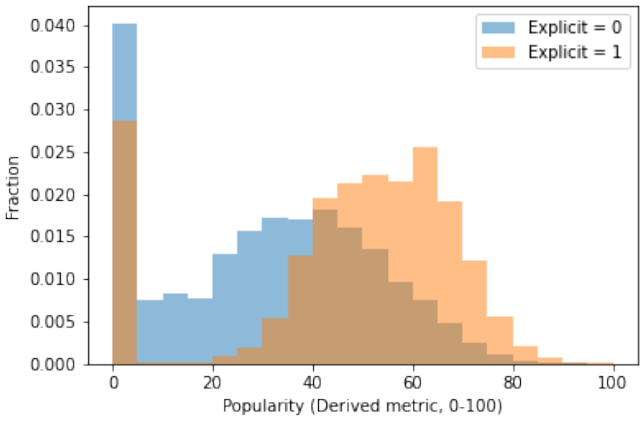

Fig.3. The histogram distribution of the fraction of popular songs for different popularity scores, for songs that contain explicit content (explicit = 1) and songs that do not contain explicit content (explicit = 0).

Fig. 4 shows the relationship between the key of a popular song and the mode of a popular song. The stacked bar plot highlights that although some keys are more likely to have more minor songs and other keys are more likely to have more major songs, overall the popular songs are major songs, irrespective of key.

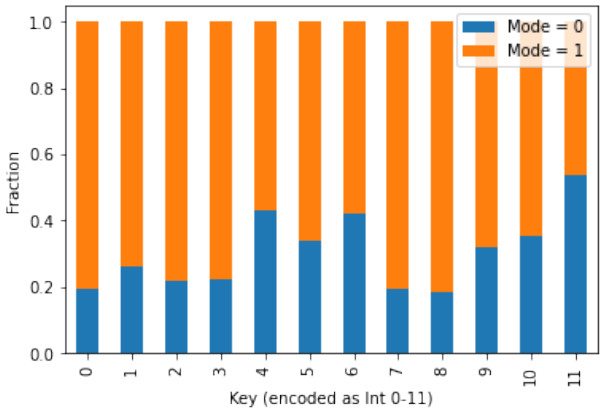

Fig.4. The bar plot of fractions of popular songs written in different keys that are encoded as integer values 0-11, where the fractions are stacked by the mode of the song. A mode of 0 represents songs written in minor chords, and a mode of 1 represents songs written in major chords.

III. Methods

A. Data Splitting and Preprocessing

It is assumed that this data is independent and identically distributed, given that all observations, in this case songs, are mutually independent from each other. There is no grouped data, or time series data, and so the basic train-test-split method is used for splitting. During splitting, 60% of the observations were first allocated to training, and then 20% of the observations were allocated to validation and testing each. Since this is a fairly large dataset, no K-fold validation was necessary, and could have ended up becoming more expensive. The train-validation-testing split is 60-20-20 because although there are a large number of observations, the model should be reproducible, and should account for a large number of songs with Popularity score 0.

Although there are 8 categorical features, only 3 of them were chosen for encoding in the preprocessing step. The other 5 features (artist, name, id, release date, year) have an extremely large proportion of unique values and so these were not encoded. Those that were encoded (explicit, key, mode) are unordered categorical variables, and so the One-Hot encoder was used to fit and transform the training sets, and transform both the validation and testing sets.

Although most of the continuous numerical features range from 0-1, and MinMaxScaler could have been used for these, there were 3 features (duration, tempo, loudness) that were significantly asymmetrically distributed, and so the Standard Scaler was used for all 8 numerical features. As a result of the preprocessing, the final preprocessed dataset has 24 features.

B. Model Selection and Training

Using this splitting and preprocessing method, six different machine learning models were trained: linear regression with L1-regularisation, linear regression with L2-regularisation, linear regression with ElasticNet-regularisation, a random forest regressor, a K-nearest neighbours regressor, and an XGBoost regressor.

The R-squared score, or the coefficient of determination, was used as an evaluation metric for all six models, because of its high interpretability. All six models were hyper-parameter tuned by iterating over a grid to find the optimal parameter combination for each model, such that it resulted in the best validation-set R-squared score. These parameters and values, tried for each model, are summarised in Table 1.

This process was repeated on five different random states in order to quantify the uncertainty in the model R-squared score due to the randomised nature of some of the model training algorithms. These scores were then averaged, using the standard deviation as a method of measuring the uncertainty due to such deterministic methods.

Algorithm	Parameters
L1	**C**: 1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3
L2	**C**: 1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3
ElasticNet	**C**: 0.001, 0.01, 0.03, 0.05, 0.1; **l1_ratio**: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8
RF	**max_depth**: 3, 7, 11, 15; **max_features**: 0.5,0.75,1.0
KNN	**n_neighbors**: 1, 3, 10, 30; **weights:** uniform, distance
XGBoost	**max_depths**: 1, 10, 30, 50

Table 1: Hyper-parameters for each ML Algorithm

IV. Results

A. Model Accuracy

The R-squared score, by definition, is 0 for the baseline model. The machine learning algorithms performed better than the baseline models, with all of their scores being more than 100 standard deviations from the baseline score. Specifically, the linear regression models all had scores about 120 standard deviations above the baseline, whereas the random forest regressor, K-nearest neighbours regressor, and XGBoost regressor had scores closer to 200 standard deviations above the baseline score. The mean test scores for each model, along with their standard deviations, are shown in Figure 5.

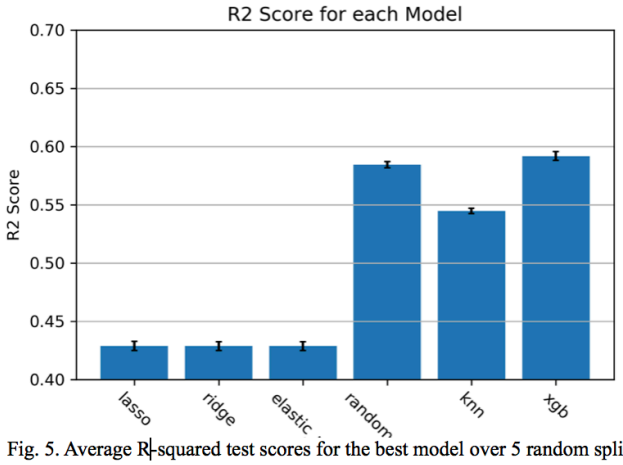

Fig. 5. Average R-squared test scores for the best model over 5 random splits

The XGBoost regressor model was the most predictive model, since it had the largest test-set R-squared score. It is thus the model of choice as the final model. After searching over the grid of parameters as well as the random states, a maximum depth of 10 was chosen for this model, since it achieved the maximum R-squared score on the validation sets during all five random states. This model returns a test-set R-squared score of about 0.59.

B. Interpretation of Findings

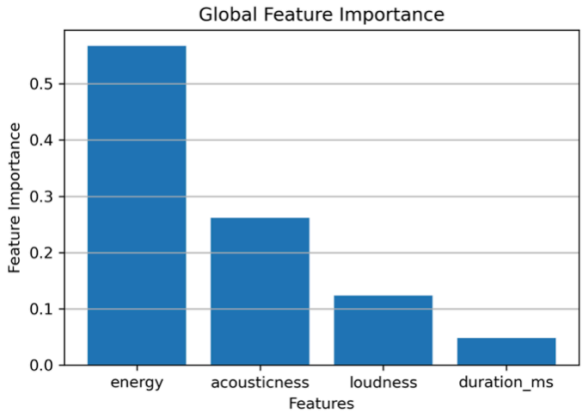

Fig. 6. Global feature importance for XGBoost regressor's best model

In order to interpret the XGBoost regressor model, global feature importance was calculated using three different methods. First, a permutation test was conducted over ten shuffles and the resulting effect on the scores was assessed to see the impact of that feature on the model. Second, XGBoost's feature importance function was used, whose results are shown in Figure 6, which shows the average feature importances of features with non-zero values. Third, standardised coefficients of the models were used to see the absolute value of their impact on the model.

All of them yielded fairly similar results, with most features having negligible importance. The top three important features were shown to be the song's energy, acousticness, as well as loudness, in that order. To check the robustness of these results, local feature importance was also evaluated. This was done using Shapely values that were computed for randomly chosen songs in the testing set.

These results are not completely unexpected, since songs with higher energy levels or loudness tend to be trending in recent times, and thus they are likely to have a higher popularity score. In the EDA section the relation between energy and loudness was observed, which should be taken into consideration here, since they both may be impacting each other's importance scores and confounding the results. Additionally, it is interesting to see that the acousticness of a song also seems to be strongly predicting its popularity, which suggests that audiences are likely polarised by songs that can be classified as acoustic. Furthermore, what is most unexpected is that explicit_0 or explicit_1 did not show a significant effect on the popularity, which was initially hypothesised based on EDA results.

V. Outlook

While the trained XGBoost regressor that was chosen as the final model did significantly better than the other models, there is some overfitting observed in the model, which could be improved. Additionally, the test-score of approximately 0.59 can be further improved. A way to do this would be to tune the other parameters in the XGBoost regressor, such as the subsample size or the min_child_weight.

To improve the interpretability and feature importance scores, possible multicollinearity could be further explored, and if necessary adjusted for, between features that may be positively correlated such as loudness and energy.

Since a large number of songs with popularity 0 were observed during EDA, the model could also be improved if more data was collected such that it had nonzero popularity. Finally, additional algorithms and techniques that were not explored in this project could be tested, for example neural networks or support vector machines.

VI. References

[1] Ochi, Virginia et. al. "Spotify Danceability and Popularity Analysis using SAP." FEMBA, California State University Los Angeles, https://arxiv.org/pdf/2108.02370.pdf

[2] Jesperhemmingsson. "Jesperhemmingsson/Spotify-EDA: An Exploratory Data Analysis of 'Spotify Dataset 1921-2020, 160k Tracks' at Kaggle: https://www.kaggle.com/Yamaerenay/Spotify-Dataset-19212020-160k-Tracks." GitHub, https://github.com/jesperhemmingsson/Spotify-EDA/

[3] Ddhartma. "Ddhartma/Spotify-Dataset-Analysis-160kTracks-1921-2020: A Song Popularity Study of 160k Spotify Tracks between 1921-2020 via Descriptive Stats and Linear Regression." *GitHub*, https://github.com/ddhartma/Spotify-dataset-analysis-160kTracks-1921-2020.