Chart Longevity and Stardom: Predicting Song Duration and Artist Breakouts on Spotify

Marc Violides **Carlos del Campo Olano** **Harish Dhatchina Moorthy**

Abstract

This project utilises data from the top 200 playlists given to us by Spotify, spanning from 2017 to 2023, combining generic variables like artists names and nationalities with additional engagement metrics like danceability and points accumulated. We perform a dual analysis on the data: the first one consists of a regression task that aims to predict the longevity of a song in the charts, and the second a classification task aiming to predict whether an artist will be a star or a one-hit-wonder. We perform different feature engineering steps for these tasks, fit vanilla models on each and optimise the best performing models on the validation set, and lastly evaluate on test set. For regression, we get that linear regression is the best-performing model with an average error of around 3 days per song duration; whereas for classification, the improved XGBoost achieves an accuracy of 71% with a balanced target variable. Overall, our models perform well considering the limited amount of variables at our disposition compared to the literature.

1 Introduction

Music streaming platforms, such as Spotify are now the primary medium for music distribution, consumption and discovery. At the heart of their success lies their reliance on data-driven insights to understand the music and the preferences of their users, providing them with the best quality end products. In the aim of gaining further insights into the music and its industry Spotify regularly releases datasets providing data about songs in the Spotify charts. One of these is a dataset containing musical features about the Top 200 songs during a year. This dataset is invaluable for the music industry as it provides insights into how musical features affect user engagement. Previous work exploring the Hit Song Science problem has been conducted with varying features, such as audio[5] and musical features [4] [2]. In contrast, limited publications are predicting an artist's success. Kang et al. [3] make use of XGBoost to do so. Additionally, the distribution of success in the music industry is studied in [1].

While exploring the music industry one typical question that arises is what distinguishes one-hit wonders from a lasting star? In this work, we aim to delve into this question by employing machine learning models and the aforementioned dataset to try to predict whether an artist is going to be a long-lasting star or a fleeting one-hit wonder and investigate if an artist's longevity is related to the music he produces.

Additionally, the life cycle of a song from its release to its peak and decline (or sustained success), involves numerous features like its genre, marketing strategies, social media or musicality. Analyzing the available dataset from Spotify, we aim to identify features of songs experimenting success during an extended time. Furthermore, we want to predict the duration of a song in the Top 200 Spotify charts, by using machine learning models. This would provide valuable insights for the music industry and Spotify itself.

2 Regression Task

In this regression task, we want to predict how long a song will stay in Spotify's top 200 charts. The implementation of this task proves to be very intricate since we have to account for the fact that a song can make different entries into the chart, thus leading to data redundancy, and the other issue is that the same song can appear with different ranks (or points) since the rank of a song can change daily. To resolve the first issue, we resort to selecting the period in which the song first appeared continuously in the chart, allowing for interruptions of up to 3 days. This way, we can capture the song's main tenure in the charts while accounting for the inherent volatility of music charts. As for the second problem, we average the rank so that each song has an average rank. Consequently, each row contains a different song with its unique attributes.

2.1 Preprocessing steps

As mentioned in the previous section, we enrich our dataset by aggregating features like the average rank of a song, and on top of that, we add variables representing the number of artists participating in the song and the number of different nationalities represented. Our target variable "duration" will be the duration for which the song lasted during its first appearance in the charts. After that, we remove redundant variables that do not provide us with any additional information and we are left with 9 independent variables, which are: Danceability, Energy, Loudness, Speechiness, Acousticness, Valence, avg_song_rank, Num_artists and num_nationalities. Notably, we removed Instrumentalness because it predominantly had 0 values. We then perform outlier detection on these remaining variables and remove the outliers using the IQR method, which is robust and resistant to extreme values. We then notice that our target feature is skewed to the left, which violates one of the assumptions of linear regression, so we apply log transformation for the target variable so that it follows a Gaussian distribution. Lastly, we go with a train/validation/test split of 75/15/15 to ensure adequate representation of our data at each stage, and the time-series nature of our data is taken into account in order to avoid future leakage. To do that, we sorted the songs based on the date they first appeared and ensured that those in the test set appeared after those in the train set.

2.2 Exploratory Data Analysis

This section gives us a clearer idea of what each variable in our dataset represents and the relationship between independent and dependent variables will guide us on what to expect from our models. It must be noted that there were no null values and outliers were removed using the IQR method.

2.2.1 Univariate analysis

Starting with the univariate analysis, which involves analysing each variable separately. The distributions for the individual variables are depicted in the appendix in B.

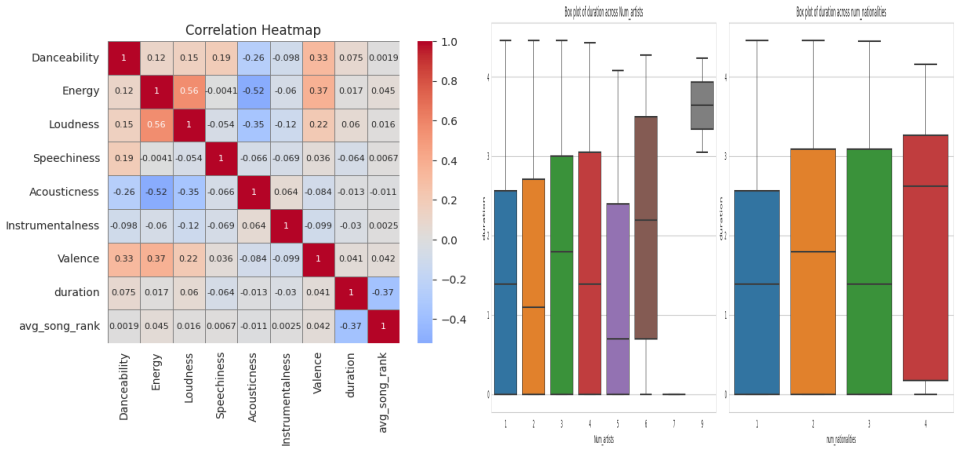
- *num_artists*: Most songs involve a single artist, with few having more than 2.
- *num_nationalities*: Most songs come from a single nationality singer, but this may be related to the fact that most have only one artist.
- *Danceability and Energy*: Both distributions appear to be fairly normal, with peak around 0.6-0.8, suggesting a moderate-to-high danceability and energy for the majority of songs
- *Loudness*: This follows a bimodal distribution, with first peak observed at around -5000db, and second peak between -60 and 0. With real data, the second peak makes more sense.
- *Speechiness*: The distribution is right skewed, suggesting that most songs have a low speechiness, so pure speech tracks (like spoken words) are rare in our charts
- *Instrumentalness*: Only contains zero values, so we will remove it during our analysis.
- *avg_song_rank*: Most songs occupy an average position in the lower part of the charts.
- *first_appearance*: Uniform distribution of songs between 2017 and 2023, suggesting a relatively balanced time-series dataset.
- ***Duration***: As for the target variable, we notice a right skewness, violating one of the assumptions of linear regression. To deal with this, we apply log transformation to make the

distribution more gaussian-like. Boxcox transformation achieved better results, but for the sake of ease of interpretability, we proceed with the former transformation.

2.2.2 Bivariate Analysis

Let's understand the relationship between the independent variables themselves and with the target.

- ***Duration vs other features***:
  - *With respect to the continuous features:* Figure 1a shows that the only moderate correlation of an independent variable with duration is that of average song rank (-0.4), which suggests that as the average rank of a song improves, its duration in the charts tends to increase. As for the other features, values close to 0 suggest that they might not linearly influence the duration a song stays in the charts.
  - *With respect to discrete features:* Figure 1b suggests that with respect to the number of artists, the variation in duration tends to follow a decreasing duration from 1 to 5 artists, but then an increasing trend is seen when going from 5 artists to 9 artists. Nothing can be concluded from this, however, since having 9 artists on a single song is a rare occurrence. As for the number of nationalities, no discernible pattern can be observed for the fluctuations of the duration, except that it increases when going from 3 to 4 nationalities.
- **Multicollinearity**: As explained in Appendix A when checking linear regression assumptions, multicollinearity is not an issue for this task, and it can be detected in Figure 1a, as no independent variables are highly correlated with each other.



(a) Correlation Heatmap for regression (b) Boxplot for bivariate analysis

Figure 1: Combined bivariate plots for regression

2.3 Regression models implementation

Before applying linear regression, we check that none of its assumptions are violated (for more details, see section A of the appendix). We proceed by fitting a wide array of VANILLA models that can tackle the regression task, mainly linear regression, decision trees, random forests, support vector machines, K-nearest neighbours (KNN) and neural networks. We selected Linear Regression because of its interpretability and simplicity, considering the limited amount of variables in our dataset. Decision Trees and Random Forests are used based on their strength in dealing with complex, non-linear relationships, which are suggested by the low correlations we saw in the bivariate analysis. SVM is included for its versatility in modelling both linear and non-linear trends, which seems to be the case for our dataset. Neural Networks are employed for their proficiency in deciphering complex patterns. Additionally, both have worked with a similar dataset [2]. KNN is chosen for its straightforward approach to understanding intricate music preferences. To evaluate these models, we use RMSE and MAE, which both measure the magnitude of error in predictions, but where RMSE penalises larger

error more so than MAE does. Employing these two metrics will give us a comprehensive overview of the results. Depending on what results we get, we want to optimise our best performing models in order to try and achieve better performance. To fine-tune the hyperparameters, we will employ grid search, in which a predefined set of hyperparameter values is exhaustively tried out with the aim of finding which hyperparameter combination provides the best results for the metrics in question.

2.4 Results

The results on the validation set of the plain model comparison are summarised in Table 1. Surprisingly, even though it is very simplistic, linear regression is the best performing model with regards to both RMSE and MAE. However, there are no hyperparameters that can be tuned and we have a limited number of variables already so we will not entertain feature selection techniques like LASSO or backward and forward selection. We will focus on hyperparameter tuning the random forest and neural network models to see if we can improve our results.

Having optimised the random forest and neural network models by tuning their most common hyperparameters through grid-search, we get the results we observe in Table 1. We notice that linear regression still outperforms the optimised models in both RMSE and MAE, making it the best fit for our regression task. This means that on average, our model is getting an error of error of $e^{1.11} \approx 3$ days, which is quite impressive considering the simplicity of the model and what we saw in our bivariate analysis.

Model	RMSE	MAE	Improved RMSE	Improved MAE
Linear Regression	1.34	1.11	-	-
Neural Network	1.36	1.13	1.32	1.10
Random Forest	1.35	1.13	1.34	1.13
SVM	1.37	1.10	-	-
KNN	1.44	1.18	-	-
Decision Tree	1.82	1.37	-	-

Table 1: VANILLA models for regression and optimised models

Our test set provides us with an unbiased evaluation of the model we chose in previous steps to see if it is able to achieve good performance on it. Our linear regression model achieves exceptional performance with an RMSE and MAE of 1.24 and 1.03, respectively. This means that on average our model is off with regards to the duration of a song in the charts by only 3 days.

3 Classification Task

In this classification task, we want to classify if an Artist is a 'One-hit wonder' or a 'Star' based on the data. The implementation for this, is a bit complex as the artist can make various entries into the chart with same or different song tracks. To resolve this complexity, we follow a somewhat a similar approach to what we did in regression, but this time by taking the average duration of that artist in the chart and keeping track of how many times they appeared in the charts. This approach allows us to set conditions upon which an artist would be considered a star.

3.1 Preprocessing steps

We add some more features to our dataset that are reflective of an artist's chart performance and musical attributes. Specifically, we included the number of entries an artist had and the average of danceability energy, instrumentalness, valence, acousticness with respect to an artist's songs. For our target variable "status", two criteria were established for an artist to be considered a star: the artist should have more than 2 entries in the charts and their average chart duration must exceed the first quantile of avg_duration for all artists, to indicate that all the songs were hits. This combination was employed to ensure a balanced and representative dataset, also fitting the description of what a star would be without being too strict. After that, we remove variables that we used to find target variable and variables that do not provide us with any additional information and we are left with 7 independent variables, which are: avg_danceability, avg_energy, avg_Speechiness, avg_Acousticness,

avg_instrumentalness, avg_Valence, Status which is our target variable. What is more, we one hot encode continent as it provides us with valuable information, as we can see in the bivariate analysis.

3.2 Exploratory Data Analysis

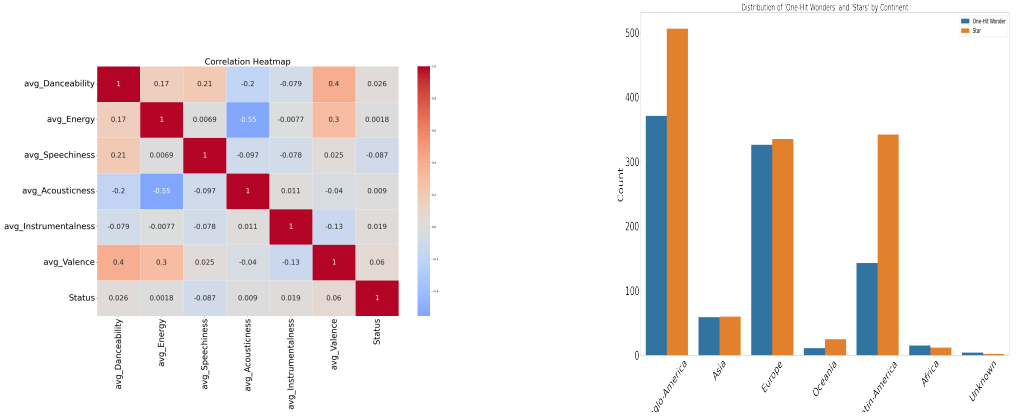
3.2.1 Univariate Analysis

We are using the same variables as the ones we discussed in the regression section, except this time we average with respect to the artist, rather than with respect to the song, but the distributions remain similar, and we additionally have the following features (visualised in Appendix C):

- *num_entries*(Number of songs) of the artist which secured a place in the charts, avg_duration (Mean duration) that an artist stayed on the chart, all his songs included, avg_Speechiness (speech) that an artist used in his songs, and avg_Acousticness (mean acoustic) of the artist with respect to all his songs, all appears to be right-skewed.
- *avg_danceability* (temptation to dance) that an artist had in his songs, and avg_energy (mean energy) of an artist displayed in his songs, appears to be slightly left-skewed.
- On the other hand, avg_Valence (rating of happiness) of an artist in his songs appeared to be equally distributed and avg_instrumentalness (instruments utilised) of an artist almost appeared flat.

3.2.2 Bivariate Analysis

When examining the correlation of an independent numerical variable with the target variable *Status* as depicted in Figure 2b, it becomes evident that there is no strong correlation between these features and the label. The connection between the label and other features appears moderately related without any significant correlation. This lack of strong correlation might be attributed to the limited data available or to a non-linear relationship. Furthermore, the analysis of *Status* in relation to categorical features, as shown in Figure 2a, reveals that the label predominantly spans across regions such as Anglo-America, Europe, and Latin America. This distribution provides insights into the geographical spread of the 'Status' variable among different categories



(a) Correlation Heatmap for classsification

(b) Categorical features for Classification

Figure 2: Combined bivariate plots for classification

3.3 Classification models implementation

For our analysis, we utilised Random Forest (RF), Logistic Regression, Neural Networks, XGBoost, and KNN models. RF models non-linear relationships suggested by low correlation between independent and dependent variables, and has proven to work well in similar contexts [4]. Logistic Regression, valued for its interpretability, effectively handles aspects of the dataset that exhibit linear relationships. XGBoost, with its high efficiency, is well-suited for the varied types of data within

Spotify, as shown in related use cases [3]. KNN's ability to model non-linear relationships makes it an excellent fit for understanding user behavior patterns in the dataset. Neural Networks, with their deep learning capabilities, are adept at uncovering intricate patterns and trends within Spotify's large data volume. Our approach involves fitting these models, optimizing the best performers based on the validation set, and then evaluating the selected model's generalization on the test set.

3.4 Results

The metrics used to analyse performance were accuracy, precision, recall and F1-score. Since our target variable is balanced, we can focus on accuracy. Results of the model comparison are summarised in Table 2. The number in brackets indicate whether we are dealing with class 0 (one hit wonder) or class 1 (star) for the target variable. It shows that Random Forest model outperforms the rest of the models without hyperparameter tuning. But after hyperparameter tuning, XGBoost classifier outperforms Random Forest model. This is to be expected, as XGBoost is a significantly complex algorithm compared to RF and involves several parameters that can alter the performance. The hyperparameters for XGBoost are set only for the first tree and the rest of the trees adjust themselves with every iteration using the *loss* of the preceding tree and carrying out gradient descent. This is beneficial in this case where the input data is expected to come in real time for Spotify and displays high variation.

Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.70	0.72[0], 0.66[1]	0.79[0], 0.57[1]	0.75[0], 0.61[1]
Neural Network	0.58	0.60[0], 0.51[1]	0.81[0], 0.27[1]	0.69[0], 0.35[1]
Improved XGBoost	**0.71**	**0.73[0], 0.67[1]**	**0.79[0], 0.60[1]**	**0.76[0], 0.63[1]**
Logistic Regression	0.55	0.57[0], 0.37[1]	0.86[0], 0.11[1]	0.69[0], 0.17[1]
KNN	0.60	0.62[0], 0.53[1]	0.79[0], 0.33[1]	0.70[0], 0.41[1]
XGBoost	0.69	0.73[0], 0.62[1]	0.72[0], 0.64[1]	0.73[0], 0.63[1]

Table 2: Models for Classification

We tested our model with a test set to evaluate its performance on unseen data. The model achieved an accuracy of 65.47%. In addition, for class *no star*, the model showed a precision of 70%, a recall of 71%, and an F1-score of 70%. This indicates an ability of the model to correctly classify instances of class *no star* in a balanced way. Furthermore, the model obtained a precision of 59% for class *star*, with a recall of 58% and an F1-score of 58%. These metrics are lower than those for class *no star*, but they still show a reasonable level of effectiveness. The similarity in the values of precision, recall and F1-scores for both classes, demonstrates a balanced performance in handling false positives and false negatives within each class. Hence, the model correctly identifies whether the artist is going to be a star or a one-hit wonder in most cases. However, the model is not perfect and miss classifies some instances.

4 Conclusion

For the regression task we achieved a very robust performance in predicting how long a song will last in the charts, as the error from the model is 3 days on average. Linear regression outperformed a variety of models, demonstrating its effectiveness, despite its simplicity.

With regards to the classification task, we observe that the impact of the independent variables is equally distributed regarding the target variable. The fact that we were able to categorize artists as either one-hit wonders or stars with 72% accuracy with such a limited amount of data and variables is remarkable and a sign of careful feature engineering and hyperparameter tuning.

Spotify can harness our insights to simultaneously pinpoint rising stars and predict their hits' chart durations, enabling smarter, dual-focused marketing efforts. To our knowledge, both of these approaches have not been explored by anyone in the related literature, so our results could serve as a baseline for future work in this domain. A potential area of expansion could involve integrating social media metrics and broader market trends to enrich the dataset and explore more complex algorithms.

References

[1] J. A. Davies. The individual success of musicians, like that of physicists, follows a stretched exponential distribution. *European Physical Journal B*, 27(4):445–447, June 2002.

[2] Elena Georgieva, Marcella Suta, and Nicholas Burton. Hitpredict: Predicting hit songs using spotify data. *STANFORD COMPUTER SCIENCE 229: MACHINE LEARNING*, 2018.

[3] Inwon Kang, Michael Manduluk, Boleslaw K. Szymanski, et al. Analyzing and predicting success in music. `https://doi.org/10.21203/rs.3.rs-1772541/v1`, June 2022. PREPRINT (Version 1) available at Research Square.

[4] Kai Middlebrook and Kian Sheik. Song hit prediction: Predicting billboard hits using spotify data. *CoRR*, abs/1908.08609, 2019.

[5] Li-Chia Yang, Szu-Yu Chou, Jen-Yu Liu, Yi-Hsuan Yang, and Yi-An Chen. Revisiting the problem of audio-based hit song prediction using convolutional neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 621–625, 2017.

Appendices

A Check for linear regression assumptions

1. **Linearity between dependent and independent variables**: points are randomly spread in the residuals vs fitted plot (see Figure 3)
2. **Normality of residuals**: most points on QQ plot follow the diagonal line (see Figure 3)
3. **Homoscedasticity:** the spread of residuals does seems to be constant across the fitted values in residuals vs fitted plot (see Figure 3)
4. **No multicollinearity:** VIF factor shows that there is a very weak correlation between the independent variables with themselves (see Table 3)

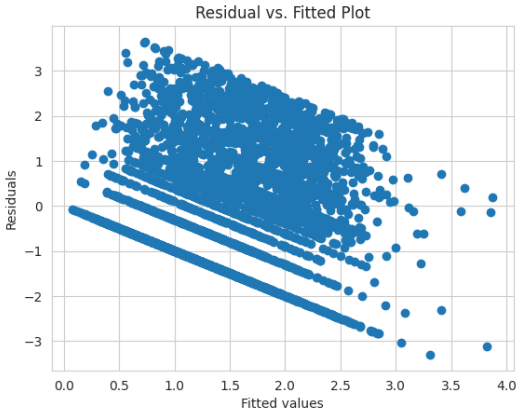


Figure 3: Residual vs fitted plot

Features	VIF Factor
Danceability	1.227
Energy	1.691
Loudness	1.282
Speechiness	1.070
Acousticness	1.244
Valence	1.320
avg_song_rank	1.014
Num_artists	1.520
num_nationalities	1.503

Table 3: VIF factors for features

B Univariate Analysis for Regression Plot

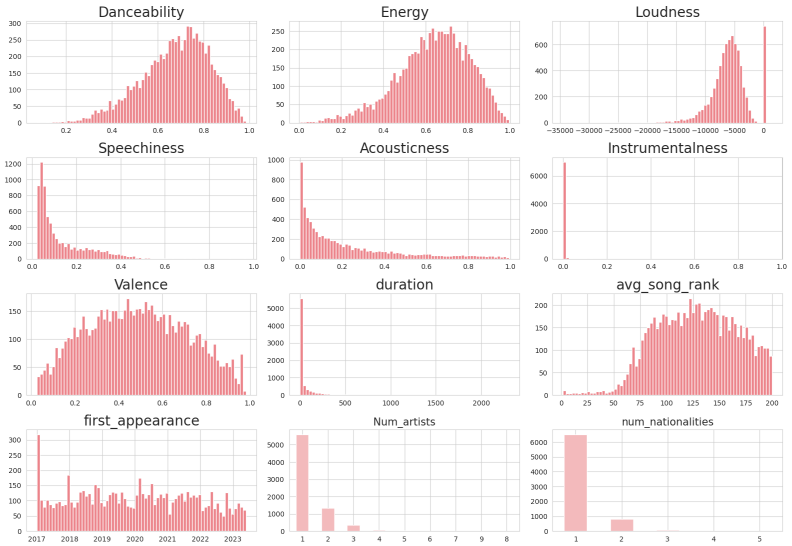


Figure 4: Combined Univariate Analysis for Regression

C Univariate Analysis for Classification Plot

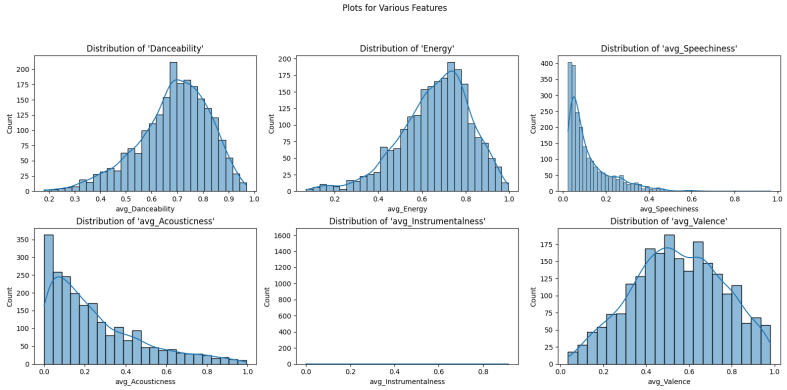


Figure 5: Univariate Analysis for Classification

D Statement of Contribution

Every member of the group contributed equally to the final report and the workload was split fairly. In addition, every member of the group contributed to each task, including EDA, preprocessing, model implementation, model evaluation and the writing of the report.

E Use of generative AI

We used generative AI within the framework and conditions tolerated by the university. Notably, we used ChatGPT to check for any grammatical inconsistencies and certain wordings of phrases, as Latex speller check demonstrated a certain inability to check for grammatical mistakes.