Song Popularity Prediction on Spotify Dataset

Course:	Probabilistic Machine Learning (SoSe 2025)
Lecturer:	Alvaro Diaz-Ruelas
Student Name:	Jonas Schwämmle
Git Hub Username:	JonasSchw14
Date:	01.08.2025
Project-ID:	08-2SJXXXX_song_popularity_spotify_kaggle

Leipzig University

Introduction

The music industry today faces the challenge of identifying the factors that contribute to a song's success in an oversaturated market with millions of tracks available. This study explores the prediction of song popularity based on audio features and metadata from the Spotify ecosystem, employing various machine learning approaches with a particular focus on Bayesian regression.

Song popularity is a complex phenomenon influenced by a variety of musical characteristics such as tempo, energy, danceability, and emotional valence. While traditional approaches to predicting success often rely on subjective evaluations, modern audio analysis technologies enable the objective quantification of musical features. The central research question of this study is: To what extent can musical audio features be used to predict song popularity, and what insights does Bayesian regression offer compared to traditional machine learning methods?

Data Loading

For this analysis, a comprehensive Spotify dataset is used, comprising both highly popular and less popular songs. The dataset includes quantitative audio features such as *energy*, *tempo*, *loudness*, *danceability*, and *valence*, as well as categorical variables like genre and musical key. Popularity is represented as a numerical score ranging from 0 to 100, based on the relative number of streams in comparison to other songs. To facilitate a comprehensive analysis, both datasets were loaded separately and subsequently merged into a single, unified dataset. The data was sourced from Kaggle and was originally generated using the Spotify API(Ameh, 2025).

Since the datasets were provided in tabular format, the loading process was straightforward. The Python library pandas was employed for this task, allowing efficient reading, manipulation, and combination of the data. This approach ensured a smooth and consistent data ingestion pipeline, serving as a solid foundation for the subsequent stages of analysis. The resulting dataset initially comprised 4,379 songs, each with 29 features.

Data Preprocessing

Data Cleaning

The initial step in the data cleaning process involved handling missing values and duplicates. A check revealed a small number of rows with missing values, which were systematically removed to ensure data integrity. Duplicates were then identified and removed based on the unique *track_id*. This step was essential to prevent identical songs from skewing the modeling results. After completing these cleaning procedures, 4,373 unique songs remained in the dataset.

Correlation Analysis

To uncover relationships between numerical audio features and to identify potential multicollinearity, a correlation matrix was computed and visualized as a heatmap. The analysis revealed a particularly strong positive correlation (greater than 0.8) between the features *energy* and *loudness*. High correlation between predictors can negatively affect the interpretability of linear models. Based on this finding, the feature *energy* was removed from the dataset to improve model stability. The correlation analysis also indicated that *loudness* and *instrumentalness* showed the strongest (positive and negative, respectively) correlations with the target variable *track_popularity*.

Feature Engineering and Selection

In the following step, features were systematically transformed and selected to maximize their predictive value for modeling. The *track_album_release_date* was used to extract the *release_year*, introducing temporal context as a numeric feature for further analysis. Encoded versions of the features *playlist_genre* and *playlist_subgenre* were generated, whereby their string values were converted into integer values, thus enabling the models to process them.

A number of features were identified as irrelevant for predicting popularity and were removed. These included unique identifiers, URLs , text-based names, and information about the type of playlist the song was featured in. This step reduced noise and results in a total number of 15 features.

Probabilistic Modeling Approach

In this project I will have a look at the following models

1. Linear Regression (Ordinary Least Squares - OLS) as a baseline.
2. Regularized Linear Models (Ridge, Lasso, Elastic Net) to handle multicollinearity and for feature selection.
3. Bayesian Regression as a probabilistic approach to quantify model and prediction uncertainty.
4. Random Forest Regressor as a non-linear ensemble model to capture complex relationships.

Linear Regression (OLS)

Linear regression serves as a fundamental baseline model. It is easy to interpret and provides a reference value against which the performance of more complex models can be measured.

As a first step, linear regression is ideal for identifying basic linear trends in the data and establishing a baseline performance. The model attempts to approximate the target variable y (popularity) as a weighted sum of the predictor variables X. The mathematical formulation is:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

Here, $\hat{y}$ is the predicted popularity, x_i are the standardized audio features, and β_i are the corresponding regression coefficients. The coefficients are determined using the method of Ordinary Least Squares (OLS), which minimizes the sum of the squared residuals $(y_i - \hat{y}_i)^2$.

Regularized Models: Ridge, Lasso, and Elastic Net

For datasets with many, potentially correlated features, as in this project, OLS models are prone to overfitting. Regularized models extend the OLS objective function by adding a penalty term to control the model's complexity.

These models are particularly suitable because the correlation analysis revealed a high correlation between some features (*energy* and *loudness*), and the large number of features after one-hot encoding makes overfitting likely.

Ridge Regression (L2 Regularization) adds a penalty term proportional to the sum of the squared coefficients. The objective function is:

$$\min \left(\sum_{i=1}^{n} (y_i - \hat{y}i)^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \right)$$

The hyperparameter α controls the strength of the regularization. Ridge shrinks the coefficients towards zero but rarely sets them exactly to zero. This is useful when many features contribute a small amount.

Lasso Regression (L1 Regularization) uses the sum of the absolute values of the coefficients as the penalty term:

$$\min \left(\sum_{i=1}^{n} (y_i - \hat{y}i)^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right)$$

The key advantage of Lasso is that it can set the coefficients of irrelevant features to exactly zero, thus performing automatic feature selection.

Elastic Net combines L1 and L2 regularization, benefiting from the advantages of both. It is particularly robust when predictors are highly correlated. The objective function is:

$$\min \left(\sum_{i=1}^{n} (y_i - \hat{y}i)^2 + \alpha\rho \sum_{j=1}^{p} |\beta_j| + \frac{\alpha(1-\rho)}{2} \sum_{j=1}^{p} \beta_j^2 \right)$$

Here, ρ controls the mix between the L1 and L2 penalties.

Probabilistic Model: Bayesian Regression

As the central model of this analysis, Bayesian Regression was implemented to go beyond mere prediction accuracy and to quantify the uncertainty of the estimates.

Instead of providing just a single point estimate for popularity, the Bayesian approach allows for the specification of a credible interval. This is valuable in practice as it indicates how confident the model is in its prediction. It allows for a probabilistic view of the model parameters and predictions. In contrast to frequentist approaches, which treat

model parameters (β) as fixed, unknown constants, Bayesian statistics treats the parameters as random variables that have a probability distribution. The process follows Bayes' theorem:

$$P(\beta|\text{Data}) = \frac{P(\text{Data}|\beta) \cdot P(\beta)}{P(\text{Data})}$$

$P(\beta)$ is the Prior Distribution: Our belief about the parameters before seeing the data. In the model, Normal distributions were chosen as weakly informative priors for the coefficients β and the intercept. $P(\text{Data}|\beta)$ is the Likelihood: The probability of observing the data given the parameters. Here, a Normal distribution was assumed for the residuals. $P(\beta|\text{Data})$ is the Posterior Distribution: Our updated belief about the parameters after considering the data. This distribution is not calculated analytically but is approximated using MCMC (Markov Chain Monte Carlo) methods, such as the NUTS sampler in PyMC. The resulting posterior distribution provides not only the most likely values for the coefficients but a full distribution from which uncertainty measures like standard deviations and credible intervals can be directly derived.

Random Forest

To capture potential non-linear relationships between the audio features and song popularity, a Random Forest Regressor was employed.

It is plausible that popularity does not depend linearly on the features. For example, a very high or very low tempo might negatively affect popularity, while a medium tempo is optimal. Random Forest can model such complex, non-linear patterns and interactions between features. A Random Forest is an ensemble method consisting of a multitude of individual decision trees. Each tree is trained on a random subset of the training data (a bootstrap sample) and a random subset of the features. This dual randomness reduces the correlation between individual trees and mitigates the risk of overfitting. The model's final prediction is the average of the predictions from all the individual trees in the "forest."

Model Training and Evaluation

All models were trained on an 80/20 split of training and test data. For Ridge, Lasso, and Elastic Net, the optimal regularization parameter alpha was determined by searching over a logarithmic range of values to minimize the Mean Squared Error (MSE) on the test data. The Random Forest model was trained with 100 trees on the training data. The probabilistic Bayesian Regression model was implemented using PyMC. Through MCMC sampling (NUTS algorithm), 1,000 samples were drawn from the posterior distribution of the model parameters to approximate their probability distributions.

Model performance was assessed using standard regression metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2).

Results

The linear models (OLS, Ridge, Lasso, Elastic Net) and the Random Forest were evaluated based on their point predictions. For each model, three diagnostic plots were generated, see Figure 1 for OLS results and Figure 2 for Random Forest results. Both plots

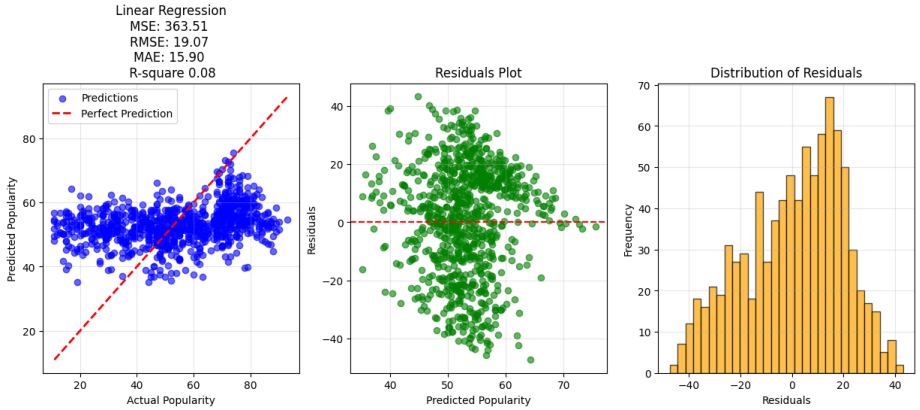

Figure 1

OLS result plots.

consist of three subplots. **Left:** scatter plot of the model's predictions against the true popularity scores. **Middle:** prediction errors (residuals) against the predicted values. For a well-behaved model, the residuals should be randomly scattered around the horizontal line at zero, showing no discernible patterns. **Right:** A histogram of the residuals. For an ideal model, this distribution should be approximately normal and centered at zero.

The evaluation of the Bayesian model focused not only on predictive accuracy but also on its primary strength: uncertainty quantification.

The predictions were generated by calculating the mean of the posterior predictive distribution for each data point in the test set. The standard metrics (MSE, RMSE, MAE, R^2) were then computed using these mean predictions. The performance was comparable

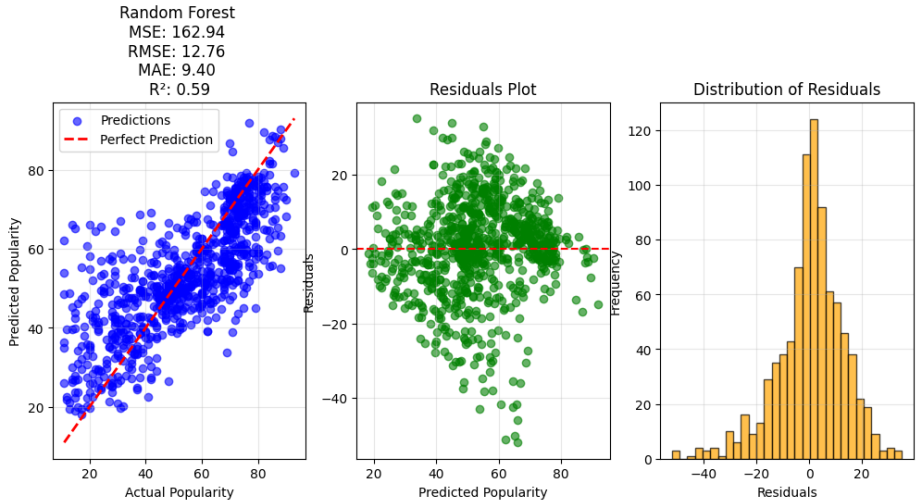

Figure 2

Random Forest result plots.

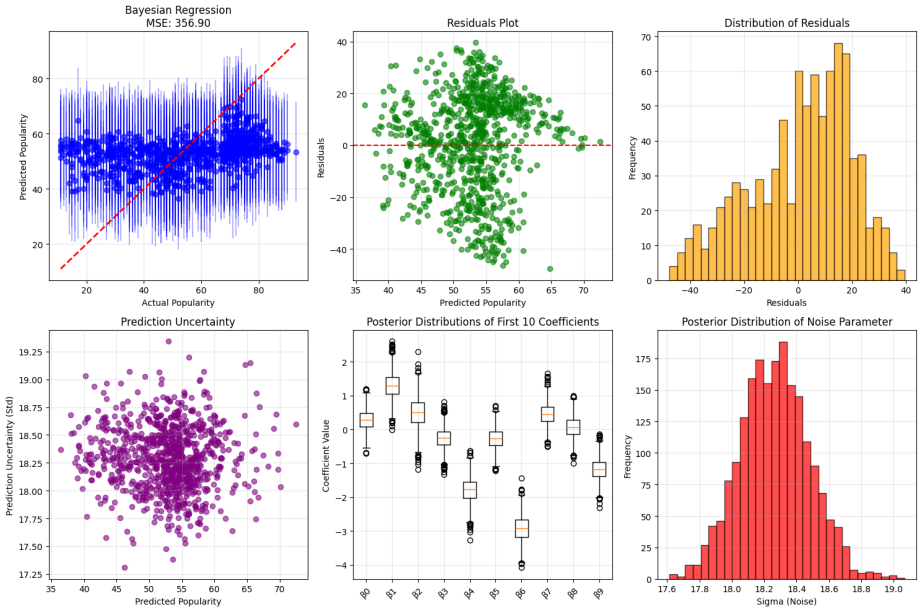

Figure 3

Bayesian regression results.

to that of the regularized linear models (Figure 3).

Key findings were:

- Superiority of the Non-linear Model: The Random Forest significantly outperforms all linear models across every metric. Its MSE is almost 20% lower, and its R^2 of 0.36 is considerably higher. This is a strong indicator that the relationship between the audio features and song popularity is complex and non-linear. Simple linear assumptions are insufficient to capture these relationships.

- Performance of Linear Models: The regularized models (Lasso, Ridge, Elastic Net) show nearly identical results to the simple linear regression. This suggests that while regularization contributed to stability, it could not overcome the fundamental limitation of the linear assumption in this case.

- Performance of the Bayesian Model: In terms of pure point predictions (mean of the posterior distribution), the Bayesian regression model performs on the same level as the other linear models. Its R^2 of 0.21 shows that, like the other linear models, it can only explain about 21% of the variance in song popularity.

Discussion

The most significant finding is the clear performance gap between the non-linear Random Forest model and all linear approaches, including the Bayesian linear regression. An R^2 of 0.36 for the Random Forest, compared to approximately 0.21 for the linear models, strongly indicates that the relationship between a song's audio features and its popularity

is not a simple linear one. This suggests that complex interactions and non-linear patterns in features like loudness, danceability, and valence are more decisive in determining a song's success than a simple weighted sum of these attributes.

Several limitations should be considered when interpreting the results. The *track_popularity* score is a proprietary Spotify metric. The exact algorithm for its calculation is not public, making it a somewhat "black box" target. It is known to be related to the number and recency of streams, but the precise weighting is unknown.

To combine the strengths of non-linearity and uncertainty quantification, more advanced probabilistic models could be implemented. A Gaussian Process Regressor would be an excellent choice, as it is non-parametric and naturally provides uncertainty estimates. Alternatively, a Bayesian Neural Network (BNN) could capture highly complex patterns while still offering a probabilistic output.

Conclusion

This project aimed to predict song popularity on Spotify by comparing various regression models, with a special focus on the value of a probabilistic approach. The analysis demonstrated that while a non-linear model like the Random Forest achieved the highest predictive accuracy ($R^2 = 0.36$), all models were fundamentally limited by the available audio features, which alone cannot capture the full complexity of what makes a song popular.

The central outcome of this project is the clear demonstration of the unique contribution of probabilistic modeling. Although the Bayesian linear regression model did not surpass the Random Forest in accuracy, it provided a crucial layer of insight by quantifying the uncertainty for each prediction. This ability to deliver not just a prediction but also a corresponding confidence level is invaluable. It provides a more transparent and realistic assessment of the model's capabilities, which is essential in a domain as noisy and multifactorial as music popularity.

References

Ameh, S. (2025). *Spotify music dataset* [accessed on 01.08.2025]. https://www.kaggle.com/datasets/solomonameh/spotify-music-dataset/data