Predicting Song Popularity with Machine Learning

Authors: CJ Phillips, Gemechu Taye, Dylan Krim

Professor: Dr. Korpusik

Abstract

This paper explores the use of machine learning to predict song popularity using Spotify's dataset. By analyzing song features such as duration, loudness, and energy, we apply and compare models like Linear Regression, Decision Trees, Random Forests, and XGBoost. While XGBoost achieves the best results with the lowest Mean Absolute Error and highest R^2 score, overfitting remains a challenge. Insights from this study suggest instrumentalness, duration, and energy significantly influence popularity, whereas features like key and mode are less impactful.

1. Introduction

Music serves as a universal medium of creativity and expression, influencing culture and emotions globally. However, predicting song success remains largely subjective, driven by trends and personal preferences. A data-driven approach could empower artists and producers to create more impactful tracks. We will be using Spotify's feature rich dataset to predict song popularity and from there Analyze patterns in song features while identifying the most influential attributes. The goal is to start the development of an interpretable model for practical application in the music industry.

2. Methodology

2.1 Data Overview

For our Data source we found a Kaggle data set that contained 32,833 tracks. Each of these tracks contained features such as track_id, track_artist, track_popularity, danceability, energy, and other metrics in terms of the song composition and data about the song like release date. We decided to split our data into an 80/20 split where we had 26,266 samples for training and 6,567 samples for testing. A StandardScaler was applied to normalize the data especially with features like keys to make it easier for the models to interpret. We implemented four different models including Linear Regression, Decision Tree Regressor, Random Forest Regressor, XGBoost. We fed our models data relating to song construction such as its tempo and dancibility.

2.2 Features

The following are the features added and how they were scored: Track popularity is given from spotify and scored on a scale from 0-100, danceability is how easy it is to dance to a song and that is given a metric of 0-1, energy is the intensity and activity of the music scored between 0-1, Key is the group of notes that form the base of the composition and each key is given a number to identify them, Loudness is how loud a song is and ranges from -60 to 0 decibels, Mode determines if the music is in a major or minor scale and is scored either 0 or 1, speechiness is a 0-1 metric that determines how much of the song is vocals, acousticness is a 0-1 confidence check determining if the song is acoustic, instrumentalness is a 0-1 confidence check determining whether the track contains vocals, Liveness and valence are two more features that describe music and are rated on a 0-1 scale, tempo is a a form of measuring music in its beats per minutes and we use the raw value of that, and lastly the duration of the song in milliseconds.

3. Results

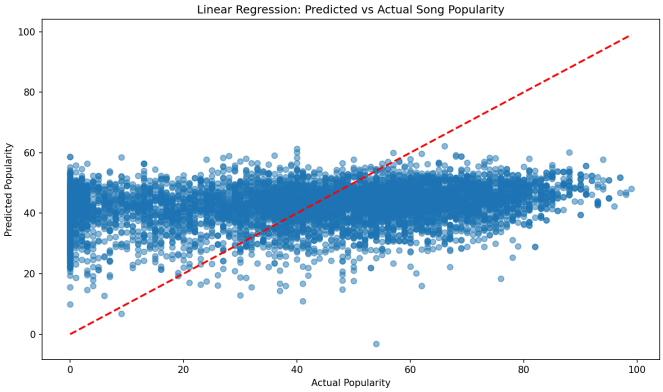
3.1 Model Results

From the four models we test here are the results

Results numerical results from the models

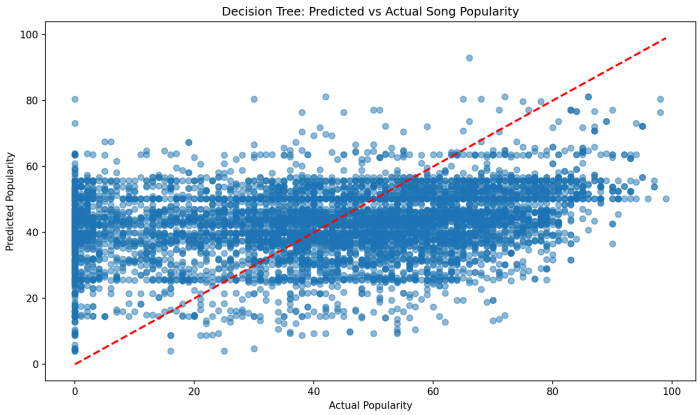
Model	Training MAE	Test MAE	Train R^2	Test R^2
Linear Regression	20.08	20.09	0.7	0.7
Decision Tree Regression	18.55	20.02	0.19	0.06
Random Forest Regression	17.86	19.25	0.27	0.15
XGBoost	12.75	18.06	0.60	0.22

Overall these all performed below what we wanted since the Baseline was 4.25 for MAE. Numerically we can see both the Linear Regression and Decision Tree that they are starting to underfit and most likely are good enough to handle the complex relation between all the different features. Our data from Random forest and XGBoost is overfitting slightly but even dealing with overfitting they will not meet the target. This most likely means we either need to clean the data more or more likely start to add more information such as song release date, artist popularity and other aspects that can help with determining popularity that are not just related to the song composition.

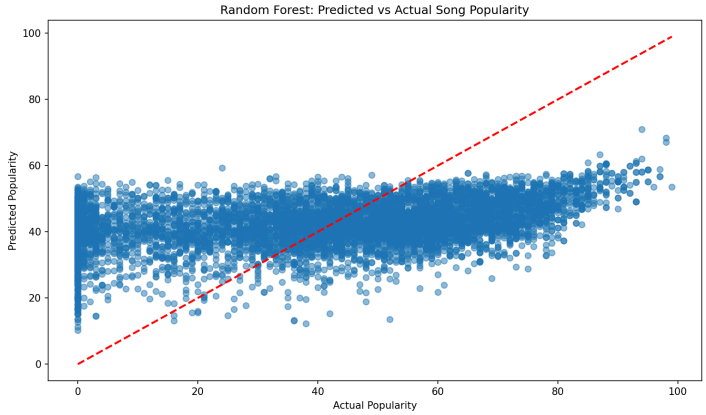


In our Linear Regression model we can see consistency but its unable to capture complex patterns making it a poor model for the task we have.

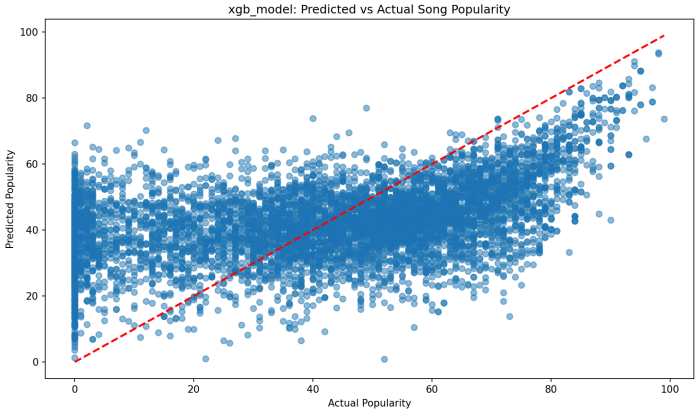
In our Decision Tree model there is Overfitting which is common in the decision tree model but it also captures noise as seen in the graph below. We believe even with multiple tweaks to the parameters this model is not capable of the task at hand as well due to how much noise it picks up when overfitting is corrected.



In our Random Forest model we can see that the model reduces overfitting but has room for improvement. Many parameters have been tested without best results currency being around a depth of 10, using n_estimator at 500 with anything above that having little to no improvement. We believe more feature engineering and possibly futuring tweaking of parameters can help improve performance.



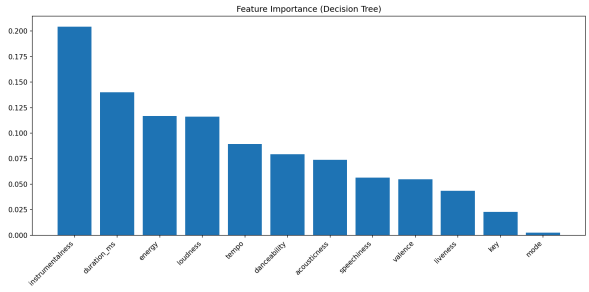
XGBoost was our most recently implemented model and currently has our best performance. The only issue with this model is that it is prone to overfitting. Even with this limitation, we believe XGBoost is the most likely model to hit our benchmark first and the extra feature engineering will help in achieving that goal.

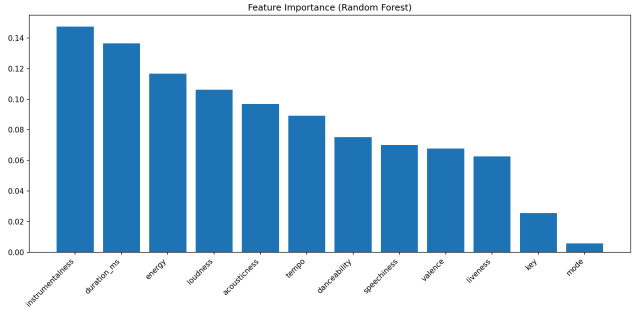


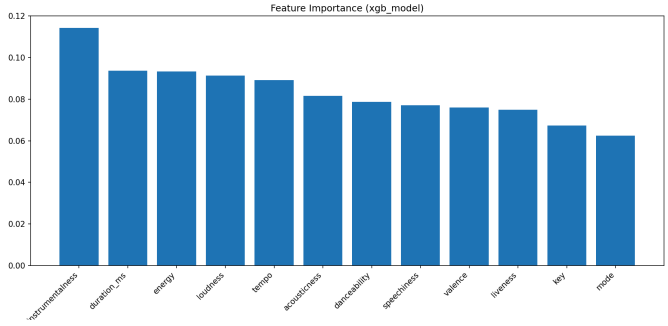
3.2 Feature Importance

From our Decision Tree, Random Forest, and XGBoost models we were also able to determine feature importance. From our results the top 4 features to determine popularity were the same with Instrumentalness being the most important. These key features were Instrumentalness, duration, energy, and loudness. The features that seem to impact popularity the least were key

and mode. These two are similar in that they determine the base of the composition of the song so we found that it makes sense that these two have little impact compared to our key features.

- 

- 

- 

Between the three models, the Decision Tree and Random Forest models have that each feature mostly declined equally in value in terms of importance. For example, Random Forest scales between 0.14 - ~0.05 and Decision tree 0.25 - ~0.00. The XGBoost model, on the other hand, had instrumentalness as higher than the others at around ~0.12 while other values only slowly declined within the range of 0.09-0.07.

4. Analysis and Discussion

4.1 Challenges

Some of the Challenges we encountered working on this project were overfitting in tree based models and difficulty in integrating complex metadata. Overfitting was rampant throughout this project but we managed to remedy the problem for the most part in our simpler models which helps us narrow down which models were best to use. The other issue mentioned was the difficulty integrating complex metadata as adding artist name directly into our data seemed to be an issue and we wanted to get stats more like artist popularity which would of required pulling from the Spotify api. Other data such as track name might not of been helpful. We also have the visualizations but those are harder to interpret, especially our preliminary views to check relations with features and popularity where graphs provided little to no information.

4.2 Future Work

We hope to add more to the project starting with more parameter tuning in order to reduce the overfitting present in most of the models. After that, we want to check if combining features would present better results so checking the relation of features to features related to popularity first. Along with those additions we would like to integrate the metadata as we feel the information will help with feature engineering and help to provide more context and hopefully better results. Lastly, improving visualization for clarity would make understanding and displaying the results better.

6. Conclusion

Our findings demonstrate that song attributes like instrumentalness, duration, and energy are predictive of popularity, though external factors remain significant. Models such as Random

Forest and XGBoost show potential but require fine-tuning to avoid overfitting and capture meaningful relationships in data.

7. Acknowledgments

Data Source: Spotify Dataset by Joakim Arvidsson.

Libraries: NumPy, Matplotlib, Scikit-learn, XGBoost.