Veritas AI Tuning into Trends: Machine Learning Models for Song Popularity Prediction on Spotify

Varun Sardana

Received February 01, 2024
Accepted March 21, 2024
Electronic access April 15, 2024

The rapid evolution of the music industry and the prevalence of digital platforms for music consumption underscore the significance of predicting song popularity. This research aims to construct a high-accuracy predictive model for song popularity by exploring diverse machine learning models and neural network architectures. Understanding the benefits of such models is important, as they offer crucial insights into audience engagement and trends. Incorporating popularity models into music recommendation systems enhances user experiences by delivering more precise and personalized song suggestions aligned with current trends. Dynamic playlist curation ensures prominent showcasing of popular and trending songs, benefiting both users and streaming platforms. For independent artists, the model serves as a strategic guide for optimal music tuning and facilitates experimentation with audience-attracting elements within their genre. Likewise, music labels can leverage predictive models to assess potential signings, guide negotiations, and inform decisions about collaborations. Our paper explores the intricacies of modeling song popularity levels and communicates the best performing model architecture we found for this application.

Keywords: Song popularity, Popularity level prediction, Spotify Song Popularity

Introduction

The music industry, a complex ecosystem intertwining artists, producers, distributors, and digital platforms, has undergone an evolution through technological advancements. With an estimated value of $26.2 billion in 2023[1], the global music landscape is witnessing a surge in various sectors. Streaming subscriptions, amassing $12.7 billion and accounting for $17.5 billion in total streaming revenues, reflect a burgeoning digital era reshaping how music is consumed and shared. This evolution is exemplified by Spotify[2], a digital powerhouse amassing a staggering 574 million users and introducing 60,000 new tracks daily into its repository—underscoring a new track every 1.4 seconds among its library of over 90 million.

This study aims to explore the phenomenon of artists striving for widespread popularity with their music. Specifically, we focus on the popularity scores assigned to songs published on platforms like Spotify, which remain determined by a complex, undisclosed equation. We propose that a song's popularity is rooted in certain inherent characteristics that resonate with listeners. By understanding and modeling these traits, artists could potentially refine their music before its public release. Our research involves analyzing a dataset containing approximately 100,000 songs, each annotated with its popularity score and various features such as acousticness, energy, and tempo. We aim to develop a predictive model using these features to estimate a song's potential popularity level. Ultimately, our goal is to provide valuable insights into the factors influencing musical appeal, enabling artists to optimize their creative output.

In our work, we focus on exploring the performance of a range of various types of machine learning models to predict the popularity of Spotify songs. In this research, we employ a variety of machine learning models to identify the most effective approach for predicting song popularity. Our exploration encompasses a comprehensive data exploration analysis along with the building of both regression and classification models. We choose machine learning models that are most applicable to the style of data that we have. We then train multiple machine learning models and compare their effectiveness to determine the highest performing architecture. We also explore model optimization techniques through hyperparameter tuning in order to identify the most accurate predictive model. Through our methodology, we aim to uncover a high performing model that can predict song popularity.

This research is aimed at constructing a predictive model tailored for song popularity assessment—a model poised to revolutionize decision-making and music generation across the industry. This predictive tool offers numerous advantages. It equips artists and labels with data-driven insights crucial for informed choices in song releases, marketing strategies, and resource allocation. The model will help the music industry optimize their resource utilization by allowing them to pinpoint songs that are likely to achieve higher popularity, thereby maximizing returns on investment. Additionally, it provides a com-

petitive edge, enabling adaptation to evolving music trends and consumer preferences.

There has been previous work that has focused on predicting model popularity. Lee, J. et al.[3] explore the characteristics of music that impact its popularity level. Specifically, this paper focuses on eight popularity metrics, and they build classification models to evaluate how accurately they can predict the popularity of a song. Araujo, C. et al.[4] present a methodology to predict if a song will make the popularity charts. Their approach attains an AUC metric ranging from .6 to.8, depending on the classifier they used. They attempted various classifiers, such as an SVM classifier with an RBF kernel. This work was focused on data provided by Spotify. Similarly, our work also focuses on a dataset of popular songs retrieved from Spotify. In our work, we present our own methodology towards data augmentation in order to attain the most accurate model to classify if a song will be popular or not.

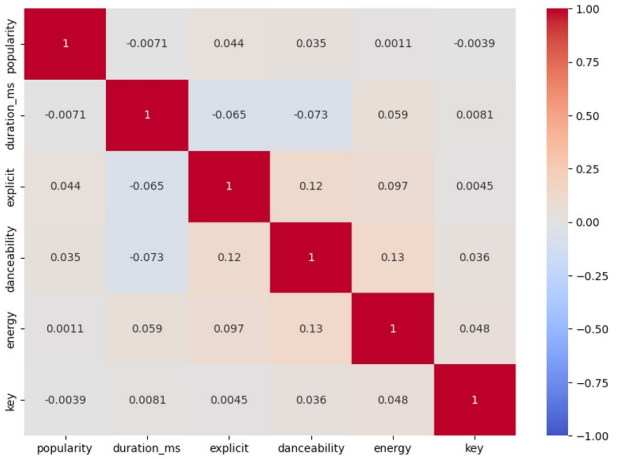

Fig. 1 A correlation matrix displaying a very low correlation between any of our features and the target popularity. A subset of features is displayed here to preserve space, but this analysis was conducted over all numeric columns available in our dataset.

Methodology

Our dataset provides a comprehensive insight into tracks on Spotify. Details about the artists performing the track, the album it belongs to, and the track's name are also provided. One of the pivotal attributes is 'popularity', calculated through an algorithm considering recent plays and the total number of plays. The dataset also explores the musical attributes of a track. These attributes include the track's duration, explicitness, danceability, energy level, key, loudness, mode, and speechiness. Furthermore, it measures how acoustic or instrumental a track is and the likelihood of it being a live recording. 'Valence' gives an idea of the track's emotional mood, while 'tempo' provides its speed in BPM. 'Time_signature' indicates the number of beats in each bar, offering a glimpse into the track's rhythmical structure. Lastly, each track is categorized into a specific 'track genre'. Overall, this dataset offers a holistic view of tracks along with features describing their key characteristics, blending both their categorical identities and intrinsic musical characteristics.

There are 114,000 entries in our dataset, each of which is a different musical entity found on Spotify. These entries precisely document data such as beats per minute, tempo, and energy levels, providing a nuanced glimpse into the core of each song. Our exploratory analysis revealed a tapestry of data types woven over our dataset's 21 columns, exhibiting a combination of four string columns and seventeen number columns. Notably, our dataset is clean, with no null values, confirming the completeness of the information included within each entry. During our exploratory data analysis, we found 450 duplicate items which we removed from the dataset for better model performance.

In our exploration of regression models for predicting the continuous numerical popularity levels of Spotify songs, we employed diverse methodologies. Specifically, our study encompassed the training and evaluation of both linear regression and MLP (Multi-Layer Perceptron) models. The primary objective of these models was to forecast a continuous numerical value representing song popularity levels within the dataset. Throughout our experimentation, we explored various architectural configurations for these models. Additionally, we tested different sets of features to optimize the predictive capacity of these regression models.

When initially delving into our regression modeling endeavors, we encountered significant challenges that resulted in the suboptimal performance of all our regression models, characterized by notably high Mean Squared Error (MSE). We identified three predominant issues during this phase of exploration, contributing to the initial unsatisfactory outcomes observed across our regression models. These challenges hindered the accurate prediction of song popularity levels and necessitated a deeper investigation into the underlying factors affecting model performance.

Linear Regression

In our Linear Regression model, we conducted a comprehensive exploration of various features, ultimately selecting Danceability, Acousticness, Speechiness, Energy, and Tempo for prediction. Our feature selection process involved analyzing the correlation matrix, revealing minimal correlation coefficients among these features. To optimize model performance, we experimented with multiple random feature combinations. We explored optimizing the performance of the linear regression model by training a set of different models, each with different features. Since there were no obvious highly correlated features, we decided to utilize a random selection of features. In addition

to this, we also experimented with applying transformations to our features such as standardization and normalization. However, despite these efforts, the model's performance remained suboptimal. Both the training and testing phases resulted in notably high Mean Squared Error (MSE) values: 495.31 for training and 495.12 for testing. Despite the diversity in feature selection attempts, the model's inability to accurately predict song popularity levels persisted, indicating limitations in its predictive capacity within the regression framework.

MLP for Regression

Since the linear regression model was not complex enough to capture all the relationships between the features and our target, we explored the use of a neural network in hopes of modeling a more complex function. In this section, we describe the use of a Multilayer Perceptron (MLP) model for regression. The selected features for this model included Danceability, Acousticness, Speechiness, Energy, Tempo, Loudness, Mode, Instrumentalness, Liveness, Valence, Time Signature, and Key. To handle discrete features like Key, Time Signature, and Mode, we employed dummy encoding. Additionally, we standardized all features to ensure consistent scaling for model training.

The neural network architecture consisted of multiple dense layers with Rectified Linear Unit (ReLU) activation functions, batch normalization, and dropout layers with a 20% dropout rate to prevent overfitting. The architecture can be summarized as follows:

MLP Architecture
Input Layer (Dense 512)
Hidden Layer 1 (Dense 256)
Hidden Layer 2 (Dense 128)
Hidden Layer 3 (Dense 64)
Hidden Layer 4 (Dense 32)
Hidden Layer 5 (Dense 16)
Hidden Layer 6 (Dense 8)
Output Layer (Dense 1)

Table 1 An Illustration of Our Constructed MLP Architecture Featuring Dense Layers with Gradually Decreasing Node Counts towards the Network Output.

In terms of performance evaluation, we measured the Test Mean Squared Error (MSE), which yielded a value of 443.93. This approach utilized an extensive set of features and applied meticulous feature engineering techniques to configure the MLP model for regression. Since our MLP performance was found to be low, we conducted an exploration of a set of different MLP architectures. This involved testing various numbers of layers, nodes, and types of activation functions. Despite these efforts, the achieved Test MSE, while showing improvement compared to previous attempts, suggests that further refinement may be required to enhance the model's predictive accuracy for song popularity regression. As displayed in Figure 2, the MLP model does a poor job at modeling the distribution of our data.

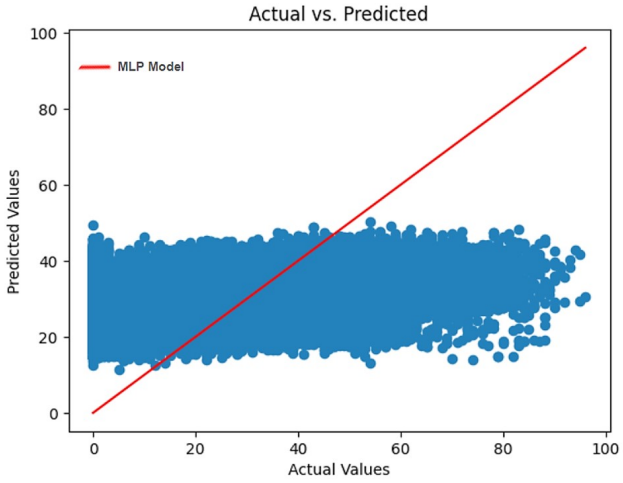

Fig. 2 Predicted values remain between the levels of 15 and 45, which shows that regression models such as our MLP could not learn the distribution of our data well.

Results

Our exploration revealed three main challenges in modeling song popularity using machine learning techniques. First, we discover a low correlation between the features used and the target popularity levels, with analysis showing minimal relationships that hinder our models' predictive accuracy. Second, our dataset is skewed by a large number of unpopular songs, making it difficult to model popularity accurately due to the overrepresentation of lower popularity songs. Lastly, there is a noticeable scarcity of data on very popular songs, which complicates understanding and predicting the factors that contribute to high levels of song popularity. Together, these challenges present significant obstacles in developing an effective model for predicting song popularity.

Problem 1: Low Correlation

Despite exploring diverse architectures within the MLP (Multi-Layer Perceptron), our efforts failed to notably alleviate the persistently high mean squared error (MSE) in our regression models. A revelation emerged during our analysis: the absence of significant correlations between the features and the specific popularity level values. In our exploration, we inspected all of our features, yet we encountered minimal correlation between these features and the target popularity levels. To visually rep-

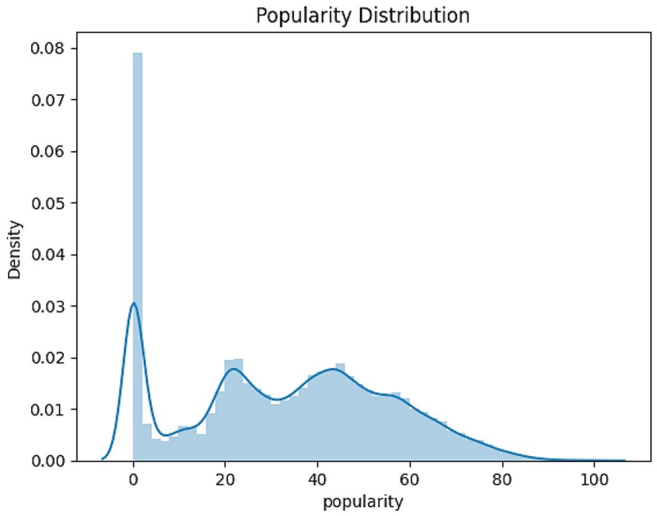

Fig. 3 Density distribution plot of song popularity in our dataset showing an unbalanced number of unpopular songs skewing our distribution.

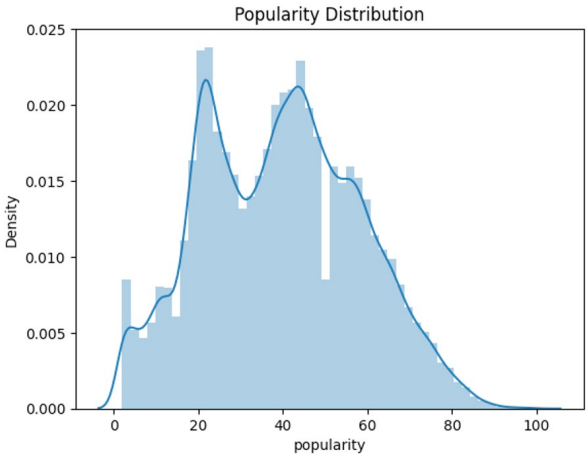

Fig. 4 A density distribution of song popularity levels in our dataset once the unpopular songs were removed. This distribution showcases a very limited number of samples between the popularity levels of 80 to 100.

resent this issue, we generated a correlation matrix showcasing the relationships within a subset of our features as seen in Figure 1. The matrix underscored a consistent pattern: negligible correlations across all the features with respect to the popularity levels. Consequently, this deficiency made it difficult for the model to discern and interpret meaningful relationships, hindering its capacity to accurately predict song popularity. This lack of discernible correlations posed a substantial challenge, impeding the model's ability to effectively capture and learn interactions between the features and the target variable.

Problem 2: A Large Number of Unpopular Songs Skewing our Dataset

Our dataset exhibited a striking prevalence of unpopular songs, presenting a considerable challenge in modeling song popularity accurately. This abundance of less popular songs within the dataset aligns with the common industry understanding that a vast proportion of published songs might not garner substantial listener engagement or reach a wide audience. To visually communicate this issue, Figure 3 presents the distribution of song popularity levels within our dataset. The graph showcases markedly higher peaks at lower popularity levels, indicating a considerable concentration of less popular songs.

Problem 3: Little data about very popular songs

Conversely, there is a scarcity of songs at higher popularity levels, reflecting the rarity of songs that achieve widespread popularity, as seen in Figure 4. This skewed distribution heavily weighted towards less popular songs poses a significant challenge for our model, as it faces difficulties in adequately learning patterns and nuances associated with highly popular songs due to their scarcity within the dataset.

Figure 3 above illustrates a striking absence of data points in the range of popularity levels between 80 and 100. This paucity of data points makes it exceedingly challenging to grasp the true characteristics and features that contribute to a song's very high popularity. Essentially, we observe a dearth of songs with an extremely high popularity level, which could be attributed to one of two factors. Firstly, it's possible that there are genuinely fewer songs in the Spotify database that attain such exceptional levels of popularity. Alternatively, it might be the case that our dataset, for various reasons, excludes a significant number of these highly popular songs.

In light of this limited information about songs with a high popularity level, constructing an effective regression-based model to predict or understand the precise factors influencing popularity becomes a formidable task. The lack of sufficient data points within this critical popularity range hinders the model's ability to discern the intricate elements that contribute to a song's meteoric rise in popularity.

Solution 1: Removal of Songs with Very Low Popularity Levels

To address the skewness prevalent in the original dataset, we executed a strategic data pruning process by removing songs that exhibited extremely low popularity levels. This action was undertaken to mitigate the right-skewed distribution observed in the initial dataset. By eliminating these less popular songs, we aimed to streamline the dataset, reducing the dominance of lower

popularity levels and creating a more balanced representation of song popularity within the remaining dataset.

Solution 2: Reframing as a Classification Problem

Recognizing the challenges posed by the original regression-based approach, we strategically redefined the problem as a classification task. Instead of predicting specific popularity levels, we transformed our objective to differentiate whether a song qualifies as popular or not. To effectuate this shift, we adjusted our target variable, assigning a binary classification: songs above a popularity threshold of 50 were labeled as '1' denoting popular, while those below were labeled '0' representing unpopular. This restructuring resulted in two equally sized groups—popular and unpopular songs—which facilitated a more balanced learning process.

Solution 3: Classification Models Employed

This comprehensive exploration of classification models and rigorous hyperparameter tuning allowed us to maximize the predictive potential of each model, enabling us to select the most effective configurations for accurate classification of song popularity. The following is a list of the classification models we built and evaluated:

1. Multi-Layer Perceptron (MLP) for Classification: The MLP model was one of our chosen tools for the classification task. We employed a neural network architecture designed to identify patterns and classify songs as either popular or unpopular based on their features. To adapt the network for classification, we incorporated a sigmoid layer into the final dense layer. This sigmoid activation function transformed the output into a probability value, enabling binary classification of songs into popular and unpopular categories.

2. Decision Trees: Decision trees are hierarchical structures used for classifying data by making a series of decisions based on feature values. These models partition the dataset by feature values and create a tree structure to classify instances into different categories.

3. K-Nearest Neighbor (KNN): KNN is an algorithm that classifies data by measuring similarity. It assigns a class to an instance by identifying its nearest neighbors in the feature space and selecting the majority class among those neighbors for the instance.

4. Random Forests: Random forests are an ensemble learning method that consists of multiple decision trees. They aggregate predictions from these trees to improve accuracy and reduce overfitting.

Table 2 Test Accuracy of Four Trained Machine Learning Models, Including the Highest Accuracy Achieved After Hyperparameter Tuning.

Model Type	Test Accuracy	Test Accuracy After Hyperparameter Tuning
MLP	75.4%	75.4%
Decision Tree	74.7%	75.36%
KNN Classifier	74.4%	75.36%
Random Forest Classifier	84.5%	84.7%

Solution 4: Model Optimizations

Our comprehensive exploration of classification models and rigorous hyperparameter tuning allowed us to harness the full predictive potential of each model, enabling us to select the most effective configurations for accurate classification of song popularity.

Hyperparameters play a crucial role in the performance of machine learning models. Optimizing these parameters, such as learning rates, tree depths, or the number of neighbors in KNN, directly impacts a model's performance and generalization. There is no mathematical equation to solve for the most optimal hyperparameters of a model. Within the field of machine learning, an important task is to sweep through a large selection of hyperparameters and train a model for each set of hyperparameters. These trained models are then evaluated in order to find the best performing hyperparameters. To identify the most effective configuration for each model, we conducted a systematic hyperparameter sweep. This involved an extensive search across various hyperparameter combinations for each classification model. For each model we selected a set of values for the most common hyperparameters and conducted a grid search where each combination of hyperparameters was tested. Through iterative testing of different parameter values, we aimed to optimize model performance and identify the parameter settings that produced the highest accuracy in predicting song popularity classification.

Among the various models that we tested,. The MLP model for classification, leverages the same comprehensive set of features and employs transformations to enable binary classification based on song popularity levels. The model demonstrated a test accuracy of 75.4%, indicating its capability to categorize songs into popular and unpopular categories. Further assessment of the train's accuracy will provide a complete picture of the model's performance.

In our pursuit of exploring alternative modeling approaches, we maintained consistency in our feature selection, employing the same set of features as utilized in previous models: Danceability, Acousticness, Speechiness, Energy, Tempo, Loudness, Mode, Instrumentalness, Liveness, Valence, Time_signature, and Key. Our feature engineering processes mirrored previous methods, encompassing the transformation of the target variable 'y' into binary values, dummy encoding for discrete features,

and standardization for normalization purposes.

For the Decision Tree for Classification, we delved into hyperparameter tuning endeavors, aiming to optimize the model's performance. Employing a grid search method with 3 folds for each of 100 candidates, totaling 300 fits, we identified the best hyperparameters as follows: 'min_samples_split': 2, 'min_samples_leaf': 16, 'max_depth': 10, 'criterion': 'entropy.' Initially, this approach displayed signs of overfitting, evident in the remarkably high train accuracy of 99%. Subsequently, through rigorous hyperparameter tuning, the model's test accuracy improved to 75.2%. These identified optimal hyperparameters, encompassing parameters like minimum samples for split and leaf, maximum depth, and criterion for splitting, contributed to the enhancement of the model's classification accuracy. Nevertheless, further optimization may still be possible to enhance the model's generalization performance. Additionally, we trained a KNN Classifier with the same set of features and feature engineering techniques as our previous models. Before any hyperparameter tuning, the KNN model achieved a testing accuracy of 74.4%. Lastly, we employed a Random Forest Classifier, retaining the same features and feature engineering processes. Before undergoing any hyperparameter tuning, the Random Forest Classifier demonstrated a testing accuracy of 84.5%.

Discussion

Out of all the machine learning models that we built, we observed that applying a classification methodology on our data instead of a regression style methodology helped us better model our data. Since our data was skewed towards songs that have a very low popularity, it was useful to divide up our data into two classes instead of predicting the specific popularity level.

The shift from a regression to a classification approach was primarily influenced by the skewed distribution of our data towards songs with lower popularity. The binary classification of songs into "popular" and "unpopular" categories circumvented the challenges posed by predicting a continuous popularity index, which our initial analysis showed had weak correlations with song characteristics. This approach aligns with the theory that classification can be more robust in cases where the data does not support the assumptions required for regression analysis.

The exceptional performance of the Random Forest classifier in our study, achieving an accuracy of 84.5%, is a testament to the power of ensemble learning methods in handling complex predictive tasks. Random Forest operates by creating a 'forest' of decision trees during its training phase, with each tree built from a random sample of the data and making predictions independently. These trees are then aggregated to make a final decision, typically through majority voting for classification tasks or averaging for regression tasks. This method leverages the strength of multiple learners to achieve better predictive performance and robustness than could be obtained from any of the individual models alone.

One of the key strengths of the Random Forest algorithm is its ability to reduce the risk of overfitting, one of the common pitfalls of decision tree models. Since each tree in the forest is built from a different subset of the data and considers a random subset of features at each split, the variance of the model is greatly reduced without substantially increasing the bias. This results in a model that is both accurate and generalizes well to unseen data.

Random Forest's versatility in handling different types of data makes it an ideal choice for datasets with a mix of numerical and categorical features, such as our song dataset. It inherently performs feature selection, giving higher importance to more relevant features and thus improving the model's accuracy. This capability is crucial when dealing with high-dimensional data, where irrelevant or redundant features can adversely affect model performance.

The ensemble method underlying Random Forest also contributes to its robustness in dealing with the complexity and variability of data. By aggregating the predictions of numerous trees, Random Forest mitigates the noise and captures complex relationships within the data without being swayed by outliers. This approach is particularly effective in scenarios where the data distribution is unknown or the relationship between features and the target variable is nonlinear.

We also were surprised to find that the hyperparameter tuning did not significantly increase the accuracy of our model. This reveals several critical insights. Firstly, the "lottery" nature of hyperparameter tuning emphasizes the randomness and unpredictability in finding the optimal configuration that significantly enhances model performance. Our experience suggests that the marginal improvements gained did not justify the computational cost and time investment, possibly due to already having reached a plateau of performance with the default parameters. This plateau indicates that our models, including the Random Forest classifier, might have been limited more by the nature of the data and feature selection than by the hyperparameters themselves. The challenges and limitations faced during hyperparameter tuning involve the extensive computational resources and time required to methodically explore the vast hyperparameter space. The effectiveness of tuning is also highly contingent on the selection of the hyperparameter range and the tuning algorithm used, which might not have been optimally configured in our experiments.

It is also important to note that the accuracies reported in Table 2 were evaluated on the unseen test dataset. Since the models performed well on this unseen data, it is reasonable to assume that the popularity of a new song passed into our model could be accurately predicted. However, this comes with the caveat that the features of the new songs must follow a similar

distribution to the data on which the model was trained.

Conclusions

Throughout this paper we explore a methodology to model the popularity level of Spotify songs. Through our exploratory data analysis, we reveal important aspects of our dataset which limit us from utilizing regression-based models very effectively. We then propose solutions to the dataset issues that we found which included balancing our data popularity distribution as well as reframing our problem into the classification paradigm.

For further research, there is a significant opportunity to enhance model generalizability by aggregating multiple datasets on song popularity levels. Such an approach could provide a more robust and comprehensive framework for understanding the multifaceted factors that contribute to a song's popularity. In the rapidly evolving field of generative AI, future research could take an innovative turn by leveraging the features strongly associated with song popularity. This could involve the automatic generation of new songs or the modification of existing tracks to imbue them with characteristics typical of popular songs. By integrating generative AI techniques, researchers could explore novel ways to generate popular songs, potentially paving the way for groundbreaking applications in music production and analysis.

Acknowledgements

Thank you for the guidance of Jason Jabbour from Harvard University in the development of this research paper.

References

1 ReportLinker, *GlobeNewswire News Room*.

2 C. Benitez, *20 Spotify statistics 2022: Usage, revenue more*, Tone Island.

3 J. Lee and J. Lee, *IEEE Transactions on Multimedia*, **20**, 3173–3182.

4 C. Araujo, M. Cristo and R. Giusti, 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA, p. 859–864.