Understanding Song popularity through AI

Ming Chen

Northeastern University, Boston, MA

Abstract

In this paper, we explore song popularity in the digital era using Artificial Intelligence (AI), focusing on Spotify's song data from Kaggle, which includes audio features and is labeled using the Billboard Hot 100 chart. This study redefines the challenge as a binary classification problem, employing four machine learning techniques: Logistic Regression, K-Nearest Neighbors, Random Forest, and Neural Networks, to predict a song's hit potential. Among these, Random Forest stands out with an 83.7% prediction accuracy.

Introduction

Music holds an essential role in daily life, particularly in today's world where access to one's favorite tunes is remarkably easier compared to the previous time, which often required purchasing physical CDs or records. "The global music market, valued at USD 27,898.47 million in 2022, is projected to grow at a Compound Annual Growth Rate (CAGR) of 9.74% and reach USD 48,733.85 million by 2028" (Industry Research Co, 2023). In such a big market, the creation of hit songs is a critical concern for music labels and artists alike.

Recent years have seen a significant transformation in how music is consumed, transitioning from platforms like CDs, iTunes and YouTube to a variety of streaming services. "Spotify, leading this change, has become the most popular streaming platform with over 574 million users across more than 180 markets." (Spotify, 2023).

As we delve into the realm of predicting a song's potential for success before its release and providing creative insights for music labels and artists striving for market success, we encounter a novel and increasingly popular methodology. This approach centers on analyzing the characteristics of a song using the music audio data. With digital music's growing accessibility and technological advancements, a new field of study, Music Information Retrieval (MIR), has emerged. Within this domain, Hit Song Science (HSS) has developed, defined by Pachet(2012) as an emerging field dedicated to forecasting a song's market success before its release. Lee et al.'s(2015) research supports the feasibility of predicting a song's popularity metrics significantly better than random chance, based solely on its audio signal.

Accessing song data from Spotify has become increasingly straightforward through its API, which provides comprehensive song information, including metadata and audio features like tempo, loudness, and instrumentality. We can classify songs using the Billboard Hot 100 chart criteria: if a song has appeared on the chart at least once in the past decade, it is labeled as a 'hit'; otherwise, it is considered a 'flop'. This study employs four different machine learning classification models—logistic regression, K-Nearest Neighbors, Random Forest, and Neural Networks—to address this classification challenge. Among these, Random Forest and Neural Networks have shown the most promising performance.

- Logistic Regression (LR)
- K-Nearest Neighbors (KNN)
- Random Forest (RF)
- Neural Networks (NN)

to address this classification challenge. Among these, Random Forest and Neural Networks have shown the most promising performance.

Background

Binary classification stands as a cornerstone problem within the field of machine learning, where the goal is to categorize elements into one of two groups based on feature data. In the context of predicting song popularity, binary classification is apt for distinguishing 'hits' from 'flops'. The methods employed in this study are all well-established algorithms that have been successfully applied to binary classification problems across various domains.

- **Logistic regression:** A fundamental linear model, is particularly suitable for binary classification due to its ability to provide probabilities for the class memberships. Its simplicity and interpretability make it a starting point for binary classification tasks.
- **K-Nearest Neighbors(KNN):** A non-parametric, instance-based learning algorithm that classifies new cases based on a similarity measure, often Euclidean distance. It is effective in capturing the non-linearity

in the data without making any assumptions about the underlying distribution.

- **Random Forest:** An ensemble learning method, operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. This method is known for its high accuracy, robustness to noise, and ability to handle unbalanced data.
- **Neural Networks:** Inspired by biological neural networks, are powerful computational models that consist of multiple layers of interconnected nodes or 'neurons'. They are capable of capturing complex patterns through their deep structure, making them highly adaptable to various types of data.

Related Work

In the context of AI-driven song popularity prediction, various methods beyond the chosen 4 could be considered. Support Vector Machines (SVMs) and Decision Trees are notable mentions, with SVMs excelling in high-dimensional classification and Decision Trees offering transparent decision-making processes. Deep learning models such as Convolutional Neural Networks, represent other sophisticated alternatives.

These methods were not selected for this project due to their complexity and resource demands. SVMs require extensive parameter tuning, Decision Trees are sensitive to overfitting. CNNs, while powerful for image and audio recognition, are less suited for the structured data in this study.

The employed methods strike a balance between accuracy and computational efficiency. Logistic Regression sets a probabilistic foundation, KNN simplifies local pattern recognition, Random Forest prevents overfitting through ensemble learning, and Neural Networks capture complex, non-linear relationships. This suite of models provides a robust framework for our binary classification problem, though future investigations may include a comparative analysis with these additional methods to enrich our predictive insights.

Project Description

In this study, we have framed the problem as a binary classification task using the dataset retrieved from the Spotify API, hosted on Kaggle. The objective is to predict the hit potential of a song, formalized as follows:

$$f : X \to Y$$

where:

Y is the binary target label with $Y = 1$ indicating a 'hit' and $Y = 0$ indicating a 'flop' (non-hit).

X includes a range of audio features extracted from the songs using the Spotify API.

These audio features capture the essence of the music, I will describe those features in the following section.

Dataset Description

The dataset comprises features for tracks obtained through Spotify's Web API (10k songs range from 2010-2019). These tracks are labeled as '1' (Hit) or '0' (Flop), contingent on their presence on the Billboard Hot 100 weekly chart. The features used from the dataset are as follows:

Y :

- **popularity:** The binary target variable, labeled '1' for a hit and '0' for a flop.

X :

- **danceability:** Defined as $danceability \in [0.0, 1.0]$, it assesses a track's suitability for dancing by analyzing tempo, rhythm stability, beat strength, and overall regularity.
- **energy:** Measured from 0.0 to 1.0, $energy$ quantifies the intensity and activity of a track.
- **key:** The key of the track is represented by integers according to Pitch Class notation, where 0 = C, 1 = C/D flat, 2 = D, etc.
- **loudness:** The average loudness in decibels (dB), with typical values ranging from -60 to 0 dB.
- **mode:** Indicates major or minor modality, with 1 for major and 0 for minor.
- **speechiness:** Denotes the presence of spoken words, with a spectrum ranging from more music-like ($speechiness < 0.33$) to more speech-like ($speechiness > 0.66$) recordings.
- **acousticness:** A confidence measure $acousticness \in [0.0, 1.0]$ indicating the likelihood of the track being acoustic.
- **instrumentalness:** Predicts the absence of vocals, with values closer to 1.0 suggesting a higher likelihood of the track being instrumental.
- **liveness:** Detects live audience presence, with higher values indicating a higher probability of the performance being live.
- **valence:** A measure $valence \in [0.0, 1.0]$ that describes the musical positiveness conveyed by a track.
- **tempo:** The estimated overall tempo of a track in BPM (beats per minute)
- **duration_ms:** The length of the track in milliseconds.
- **time_signature:** The overall time signature of a track, indicates the number of beats per bar.
- **chorus_hit:** The estimated start time of the chorus.
- **sections:** The number of distinct sections within a track.

Method Details

Logistic Regressions (LR): Logistic Regression is used for binary classification problems where the outcome is binary. The logistic function, also known as the sigmoid function, is used to model the probability that a given input point belongs to the class labeled as '1'.

The logistic function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Where $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$ is the linear combination of features (x) and their corresponding coefficients (β).

The probability of a data point belonging to class '1' can be written as:

$$P(y = 1|x; \beta) = \sigma(\beta^T x)$$

K-Nearest Neighbors (KNN): KNN is a non-parametric algorithm that classifies a data point based on how its neighbors are classified. I used the distance metric, Euclidean distance, to find the closest neighbors.

The Euclidean distance between two points $x^{(i)}$ and $x^{(j)}$ in an N-dimensional space is given by:

$$d(x^{(i)}, x^{(j)}) = \sqrt{\sum_{k=1}^{N} (x_k^{(i)} - x_k^{(j)})^2}$$

The class of a new data point is predicted by a majority vote of its k nearest neighbors.

Random Forest (RF): Random Forest is an ensemble learning method that builds multiple decision trees and merges their results. The classification result is typically the mode of the classes output by individual trees.

The decision for a Random Forest classifier can be represented as:

$$RF(x) = mode\{DT_1(x), DT_2(x), \ldots, DT_n(x)\}$$

where $DT_i(x)$ represents the decision of the i-th Decision Tree.

Neural Network (NN): A Neural Network consists of interconnected units or nodes arranged in layers. For the task at hand, I used a fully connected neural network with one hidden layer containing 64 neurons. Defined as:

$$Y = \sigma(W_2 \cdot ReLU(W_1 \cdot X + b_1) + b_2)$$

Where activation function in the hidden layer (ReLU)is defined as:

$$ReLU(z) = \max(0, z)$$

Given the binary classification nature of our task, the output layer comprises a single neuron with a sigmoid activation function to predict the probability that an input belongs to the positive class. The sigmoid function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The network is trained using the cross-entropy loss function, appropriate for binary classification tasks, which is given by:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

where y is the true label, $\hat{y}$ is the predicted probability, and N is the number of samples.

Experiments

Data Preprocessing

Before model construction, essential data preprocessing steps were undertaken to ensure the quality and reliability of the dataset. These steps included:

- Elimination of duplicate records to prevent data redundancy.
- Removal of instances with missing values to maintain dataset integrity.
- Extraction and utilization of data spanning from 2010 to 2019.

Cross-validation was employed across all models to ensure robustness and minimize overfitting.

Results for Logistic Regression

The performance of the logistic regression model was evaluated using the Receiver Operating Characteristic (ROC) curve, which provides insights into the model's diagnostic ability.

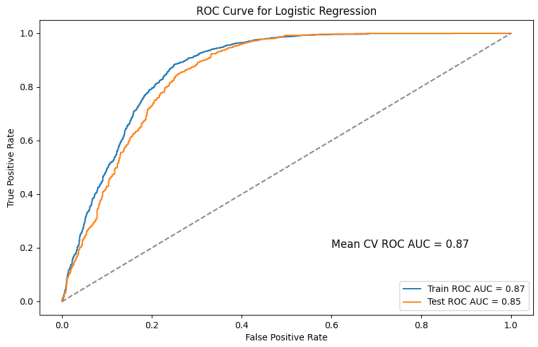

Figure 1: ROC Curve for Logistic Regression

The ideal ROC curve would hug the top left corner of the graph, indicating a high true positive rate and a low false positive rate. The proximity of the training and test curves suggests that the model is not substantially overfitting. The Area Under the Curve (AUC) on the test set was found to be 0.85, reflecting a strong discriminative ability of the model. Furthermore, the model achieved an F1 score of 0.81 on the test set, denoting a robust harmonic mean between precision and recall.

The accuracy metrics for the logistic regression model were as follows:

- Training Accuracy: 0.81
- Test Accuracy: 0.79

These results indicate that the logistic regression model exhibits commendable predictive performance and generalizes well to unseen data.

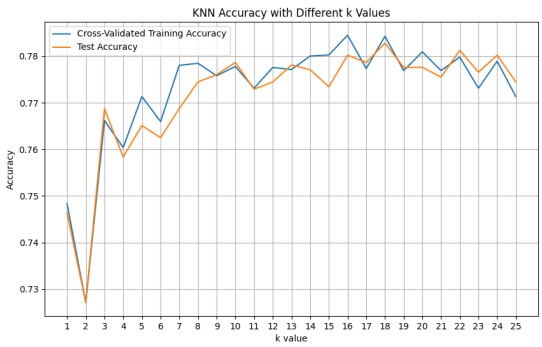

Figure 2: Accuracy of the KNN model for different values of k.

K-Nearest Neighbors (KNN)

The KNN algorithm was executed by varying the number of neighbors k from 1 to 25. The accuracies corresponding to these k values are depicted in the figure below:

Observations from the figure indicate that accuracies begin to plateau beyond $k = 6$, with both training and test accuracies leveling out without notable variance. Consequently, $k = 10$ was selected as the optimal number of neighbors for model fitting.

The model's accuracy statistics are as follows:

- Training Accuracy: 0.77
- Test Accuracy: 0.76

Random Forest

The Random Forest model was optimized using cross-validation alongside hyperparameter tuning to ascertain the optimal number of decision trees. The performance impact of varying the number of trees is illustrated in the following figure:

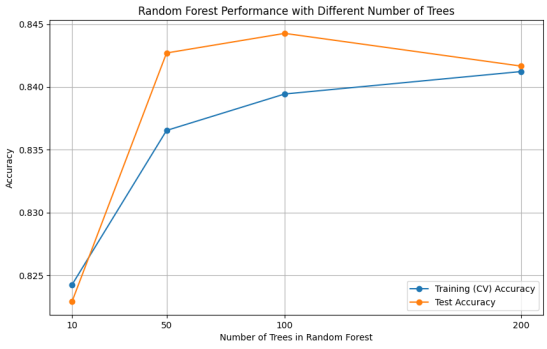

Figure 3: Random Forest accuracy across different numbers of decision trees

It was observed that the model achieves optimal performance when the number of trees, denoted as n, is set to 100.

Furthermore, the intrinsic feature importance aspect of Random Forest was leveraged to identify the most significant predictors in the dataset. The top five features, ranked by their importance scores, are:

1. Feature *instrumentalness*: Importance score of 0.2469
2. Feature *loudness*: Importance score of 0.1003
3. Feature *energy*: Importance score of 0.0949
4. Feature *acousticness*: Importance score of 0.0945
5. Feature *danceability*: Importance score of 0.0863

Neural Network

The fully connected neural network model was trained over 100 epochs. The progression of accuracy and loss over these epochs is captured in the following figures:

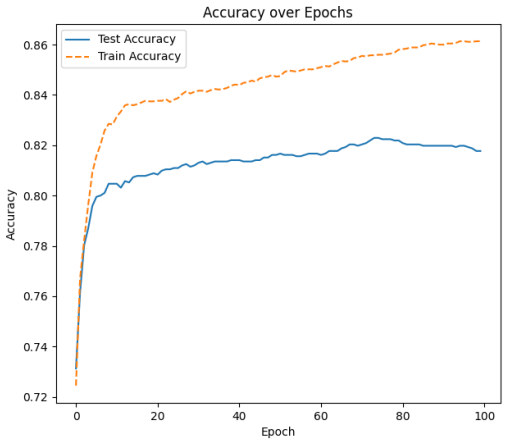

Figure 4: Model accuracy over 100 epochs

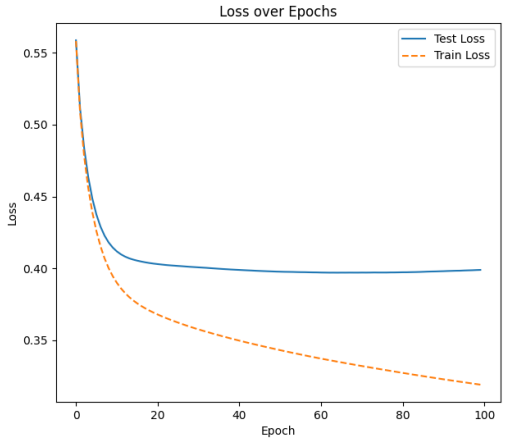

Figure 5: Model loss over 100 epochs

The analysis of the figures suggests that the model achieves the best performance at epoch 73. Continuing beyond this point appears to introduce overfitting, as indicated by the divergence of training and test accuracy.

The performance metrics for the neural network at the optimal epoch are as follows:

- Epoch with highest test accuracy: 73
- Training Accuracy: 0.8557
- Test Accuracy: 0.8229

Results Comparison

Model	LR	KNN	RF	NN
Training	81.26	77.09	83.61	85.57
Test	79.32	76.25	**83.70**	82.29

Table 1: Training and test accuracy percentages comparison

The results comparison for the four models reveals a consistently high test accuracy, ranging from 75% to 84%. Such performance is commendable and suggests that the selected audio features have a significant predictive power regarding the hit potential of a song. This level of accuracy across diverse machine learning models indicates robustness in the underlying patterns captured by the features.

Conlcusion

Our study found that machine learning can predict hit songs with high accuracy, but this doesn't mean making hits is now just a numbers game. High accuracy shows that our models can pick up patterns in music features that often lead to popular songs. However, it's not all about data – the creativity and emotions in music-making are still key and can't be captured by algorithms alone.

Predicting hits doesn't make hit-making any less complex. The music world changes fast, and what works today might not work tomorrow. Plus, making songs just to score high on a predictive model could make all music sound the same, which nobody wants.

In short, while our models are useful, they're not the whole story. The magic of making a hit song still needs that human touch, something that goes beyond what we can measure.

Reference

Web Reference
Indorf, N. (2022). Using Deep Learning to Predict Hip-Hop Popularity on Spotify. Retrieved from https://towardsdatascience.com/using-deep-learning-to-predict-hip-hop-popularity-on-spotify-1125dc734ac2

Conference Paper
Pachet, F. and Roy, P. (2008). Hit Song Science is Not Yet a Science. *Proceedings of ISMIR 2008*, 355-360, Philadelphia, USA.

Book Chapter
Pachet, F. (2011). Hit Song Science. In Tao, Tzanetakis & Ogihara (Eds.), *Music Data Mining*, Chapter X, CRC Press/Chapman Hall.

Conference Paper
Raza, A. H. and Nanath, K. (2020). Predicting a Hit Song with Machine Learning: Is There an Apriori Secret Formula? In *Proceedings of the 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, 111-116.

Journal Artical
Dimolitsas, I.; Kantarelis, S. and Fouka, A. (2023). SpotHitPy: A Study For ML-Based Song Hit Prediction Using Spotify.*arXiv preprint arXiv:2301.07978*

University Technical Report
Nijkamp, R.(2018). Prediction of product success: explaining song popularity by audio features from Spotify data.