Predicting Song Popularity by Analyzing Audio Features of Spotify Bengali Tracks across Diverse Genres

Sadia Afrin (339)
Department of CSE
Jahangirnagar University
Dhaka, Bangladesh
afrin.stu20183@juniv.edu

Tama Shil (342)
Department of CSE
Jahangirnagar University
Dhaka, Bangladesh
tama.stu2018@juniv.edu

Prokash Maitra (370)
Department of CSE
Jahangirnagar University
Dhaka, Bangladesh
prokash.stu2018@juniv.edu

Pritam Saha (375)
Department of CSE
Jahangirnagar University
Dhaka, Bangladesh
pritam.stu2018@juniv.edu

Nasim Hossain (391)
Department of CSE
Jahangirnagar University
Dhaka, Bangladesh
nasim.stu2018@juniv.edu

***Abstract*—At present, Spotify is one of the most used music streaming services and a popular song on Spotify is likely to be commercially successful. In the ever-evolving music industry, the ability to predict the potential popularity of a song before producing it can make impactful changes, leading to commercially successful music production. In this project, we present a methodology to predict if a Bengali song is going to be hit on Spotify, utilizing a robust dataset derived from the Spotify Web API. The dataset comprises various audio features of songs across multiple genres. We employed several machine learning models to forecast song popularity, specifically Logistic Regression, Random Forest, K-nearest neighbors (KNN), Gaussian Naive Bayes, Decision Tree, XGBoost, and Support Vector Machine (SVM). By analyzing data from the Spotify Web API, we aim to provide insights for more commercially successful music production in the Bengali music industry.**

***Index Terms*—song, popularity prediction, genre, music, Spotify**

I. Introduction

The music industry is very significant. According to the International Federation of the Phonographic Industry—IFPI—, the world- wide revenue from streaming, performance rights, and sales of CDs and digital music amounted to 19.1 billion USD in 2018. The majority of this figure is due to digital media: 47% of that share comes from streaming and 12% from digital sales[1]. Currently, one of the largest streaming services is Spotify, which by the end of 2018 had over 191 million active users.[2].

Spotify is a digital music service widely used at present all over the world. Bangladesh boasts a wealth of exceptionally talented singers and an impressive repertoire of melodic songs. The popularity of Bangla music transcends borders, resonating with listeners across linguistic and cultural boundaries. But we live in an age where all famous songs need not be good. Some unfitted songs are sometimes more popular than beautiful melodic songs. In this work, we predict the popularity of a Bangla song on Spotify. Our predictive model categorizes song popularity into four classes: 'Not popular', 'Kind of popular', 'Popular', and 'Very popular'. We focus on Spotify because it is widely used at present times all over the world. We have created a model that assigns a popularity class based on some attributes of the songs predicting whether the song will be famous on Spotify or not.

For this project, we created our dataset using the Bangla tracks from Spotify and used Spotify web API to generate our dataset with 17 attributes. Leveraging our curated dataset of Bangla songs across diverse genres, we employed seven machine-learning methods: Logistic Regression, Random Forest, K-nearest neighbors (KNN), Gaussian Naive Bayes, Decision Tree, XGBoost, and Support Vector Machine (SVM). Finally, we have compared the results of these methods for predicting popularity and concluded a decision on which method can be good for this prediction.

The remainder of this article is as follows: in Section II we show a literature review. In Section III we present the methodology used in our study. We described our preferred dataset in Section IV. The results we obtained are presented and analyzed in Section V. Finally, in Section VI we make a brief conclusion and point out possible future work.

II. Literature Review

In this literature review, we delve into the burgeoning field of music prediction modeling, exploring recent advancements, key methodologies, and future directions in the quest to decode the secrets of song popularity.

[1] www.ifpi.org/downloads/GMR2019.pdf

[2] https://www.fool.com/investing/2018/12/17/whats-most-popular-music-streaming-service- 2018.aspx

J. S. Gulmatico[1]., this study endeavors to construct a robust predictive model for identifying hit songs, employing a comprehensive analysis that encompasses four distinct machine learning algorithms: Linear Regression, Random Forest Classifier, and K-means Clustering. By leveraging these methodologies, the research aims to develop a prediction model capable of discerning the popularity of songs within the mainstream milieu, thereby contributing to the advancement of machine learning-based approaches for classifying songs based on their level of popularity.

Suh, Brendan Joseph,[2]., this paper analyzes Spotify API data from 2017-2018 across five countries to assess the impact of song attributes on-track success in Spotify's Top 200 Chart. Two dependent variables, peak chart position and days on the chart, are examined. Findings reveal that featuring a guest artist tends to boost a song's peak position and chart longevity across most countries. Additionally, songs perceived as "happier" exhibit greater success in Norway and Taiwan, while louder, more aggressive tracks tend to have shorter chart lifespans in three out of the five countries. The study advocates for further research with larger, more diverse datasets to validate and extend these findings to other regions.

Yee YK, Raheem M [3]., in this study, both audio features from Spotify and social media variables from YouTube music videos were utilized to predict song popularity. Random forest emerged as the top-performing model, achieving impressive accuracy (79.6%), macro-precision (74.5%), macro-recall (73.2%), and macro F1-scores (73.1%). Moreover, the incorporation of YouTube-based social media features significantly enhanced prediction accuracy, with improvements ranging from 10% to 60% across evaluation metrics. Particularly noteworthy was the 39% average increase in macro F1-scores across all models, affirming the effectiveness of combining audio and social media data for identifying potential hit songs.

C. V. Soares Araujo, M. A. Pinheiro de Cristo, and R. Giusti,[4]., This paper proposes a methodology for predicting a song's likelihood of appearing on Spotify's Top 50 Global ranking. Given the significance of online streaming platforms in music consumption, accurate prediction holds substantial financial implications for artists and labels. The approach frames the problem as a classification task, leveraging past data from Spotify's Top 50 Global ranking and acoustic features of songs. Through experimentation, the Support Vector Machine classifier with RBF kernel emerged as the most effective, achieving an AUC exceeding 80% when forecasting song popularity two months in advance.

REN, Jing, and KAUFFMAN, Robert J..[5]., the research explores the factors influencing the popularity and duration of music tracks on online streaming platforms. By analyzing over 78,000 track ranking observations from a streaming music service, the study investigates the impact of music semantics constructs (genre, mood, instrumental, theme) and non-musical factors (artist reputation, social information) on track performance. Results demonstrate the potential to predict both the duration of chart popularity and weekly track rankings. This underscores the efficacy of combining data analytics, machine-based techniques, and econometrics to uncover insights into music consumption patterns.

Rahman Autul,[6]., highlighted that Bangladeshi rock music is more popular on Spotify compared to rock from West Bengal, exploring this discrepancy through an analysis of audio features to understand cultural and musical nuances. Such insights are crucial as they underline the importance of considering regional and cultural distinctions in music appreciation studies, thereby broadening the scope of current music consumption analytics to include diverse musical landscapes.

III. Methodology

A. Data Preprocessing

As part of our data preprocessing efforts, we meticulously addressed missing values within the dataset to ensure its completeness. This involved removing records containing missing values and removing unsupportable characters. Furthermore, we converted our numerical attributes to categorical attributes to get better results. The numerical attributes that we converted to categorical are based on various sources:

1) Popularity (target class) – The popularity of the track. The value is between 0 and 100 in the raw dataset, with 100 being the most popular. The popularity is calculated by Spotify *using an unknown algorithm* based on the total number of plays the track has had and how recent those plays are. We categorized it into 4 classes.
2) Key – The key the track is in. Integers map to pitches using standard Pitch Class Notation in the raw dataset. E.g. 0 = C, 2 = D, and so on. If no key is detected, the value is -1. Range: -1 - 11. We have pre-processed the key into 3 classes using *Pitch Class Notation*.[3]
3) Danceability – Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo rhythm stability, beat strength, and overall regularity. A value of 0.0 in the raw dataset is the least danceable and 1.0 is the most danceable. We categorized it into 3 classes.[4]
4) Loudness – The overall loudness of a track in decibels(dB) in the raw dataset. Loudness values are averaged across the entire track and are useful for comparing the relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength(amplitude). Values typically range between -60 and 0 db. We assigned 3 categories to loudness according to Spotify support.[5]
5) Speechiness – Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording, the closer to 1.0 the attribute value in the raw dataset. Values above 0.66 describe tracks that are probably made entirely of spoken words. In the raw dataset, values between 0.33 and 0.66 describe

[3]https://en.wikipedia.org/wiki/Pitch$_{c}lass$

[4]https://medium.com/@agrawalananya17/spotify-song-danceability-prediction-2b999629589a

[5]https://developer.spotify.com/documentation/web-api/reference/get-audio-features

tracks that may contain both music and speech, such as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks. We assigned 3 categories in it according to information provided in Spotify Web API.[6]

6) Acousticness – In the raw dataset, acousticness is a confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic. We assigned 2 categories to it.[6].
7) Instrumentalness – Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks as vocal. In the raw dataset, the closer the value is to 1.0, the greater the likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.We categorized it into 2 classes.[6].
8) Liveness – Detects the presence of an audience in the recording. Higher liven values represent an increased probability that the track was performed live. In the raw dataset, a value of 0.8 or higher provides a strong likelihood that the track is live. We assigned 2 categories to it.[6].
9) Duration – The duration of the track in seconds. We categorized it into 3 classes.[6].
10) Energy – In the raw dataset, energy is a measure from 0.0 to 1.0 and represents a perceptual measure and intensity and activity. We have divided it into 2 categories after data pre-processing.[7].
11) Valence – In the raw dataset, a measure from 0.0 to 1.0 describes the musical positiveness conveyed by a track. Tracks with high valence sound more positive, while tracks with low valence sound more negative. We assigned 2 categories to it.[7].
12) Tempo – The overall estimated tempo of a track in beats per minute (BPM) in the raw dataset. In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration. We assigned 3 categories to it.[8].
13) Time_Signature – An estimated time signature. The time signature(meter) is a notational convention to specify how many beats are in each measure. In the raw dataset, the time signature ranges from 3 to 7 indicating time signatures of "3/4", to "7/4". We categorized it into 3 classes.[7].

B. Applying Machine Learning Algorithm

Following data preprocessing, we embarked on the core phase of our methodology: applying various Machine Learning Classification algorithms to predict song popularity. Our approach encompassed the utilization of a diverse set of algorithms, including

[6]https://developer.spotify.com/documentation/web-api/reference/get-audio-features

[7]https://courses.lumenlearning.com/suny-musicappreciationtheory/chapter/time-signature/

1) Logistic Regression
2) Random Forest
3) K-Nearest Neighbors (KNN)
4) Support Vector Machine (SVM)
5) Gaussian Naive Bayes
6) Decision Tree
7) XGBoost

Each algorithm was trained and evaluated on the preprocessed dataset to ascertain its efficacy in accurately predicting song popularity across different genres.

A brief overview of each algorithm is as follows:

Logistic Regression:

Logistic regression models the probability that a given input belongs to a certain class. It uses the logistic function (also known as the sigmoid function) to map predicted values to probabilities between 0 and 1.

$$P(y=1|x)=\frac{1}{1+e^{-z}} \tag{1}$$

where,
$P(y=1|x)$ is the probability of the output being 1 given the input x, and z is the linear combination of the input features and their corresponding weights.

The model is trained using maximum likelihood estimation or gradient descent to find the optimal weights that minimize the logistic loss function.

K-Nearest Neighbors (KNN):

KNN is a lazy learning algorithm that classifies data points based on the majority class of their k nearest neighbors in the feature space. The distance metric (e.g., Euclidean distance) is used to measure similarity.

This algorithm involves:

- Calculating the distance between the query point and all training points.
- Selecting the k nearest neighbors.
- Assigning the class label by majority voting.

There's no explicit training phase, but optimization can involve techniques like KD-trees or ball trees for efficient nearest neighbor search.

Random Forest:

Random Forest builds multiple decision trees during training and outputs the mode of the classes as the prediction. Each tree is trained on a random subset of the data and a random subset of the features, which introduces randomness and reduces overfitting.

Each decision tree splits the data based on feature thresholds to maximize information gain or Gini impurity reduction. The algorithm aggregates predictions from multiple trees, reducing variance and improving generalization. Hyperparameters like tree depth, number of trees, and feature subsampling rate can be optimized.

Support Vector Machine (SVM):

SVM finds the optimal hyperplane that separates data points into different classes with the maximum margin. It can handle linearly separable as well as non-linearly separable data using kernel tricks.

$$y = \text{sign}(w^T x + b) \quad (2)$$

where,
w is the weight vector, x is the input vector, b is the bias term, and the sign function determines the class label.

The objective is to maximize the margin between the hyperplane and the nearest data points (support vectors), while minimizing the classification error. This is typically done using convex optimization techniques.

Gaussian Naive Bayes:

Naive Bayes classifiers are based on Bayes' theorem with the assumption of independence between features. Gaussian Naive Bayes specifically assumes that features follow a Gaussian distribution.

$$P(y|x_1, x_2, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, x_2, ..., x_n)} \quad (3)$$

where,
$P(y|x_1, x_2, ..., x_n)$ is the probability of class y given the input features $x_1, x_2, ..., x_n$.

The model parameters (prior probabilities and class conditional probabilities) are estimated from the training data. Laplace smoothing or other techniques may be used to handle zero probabilities.

Decision Tree:

Decision trees recursively split the data into subsets based on the value of features to minimize impurity (e.g., Gini impurity or entropy). Each split is chosen to maximize information gain or impurity reduction. Decision trees make decisions based on conditional statements involving feature thresholds.It can overfit the training data, so techniques like pruning (removing unnecessary branches) and setting maximum depth are used to prevent overfitting.

XGBoost (Extreme Gradient Boosting):

XGBoost is an optimized implementation of gradient boosting, which sequentially builds multiple decision trees to correct errors made by previous models. It uses a gradient descent optimization algorithm to minimize a predefined loss function.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad (4)$$

where,
$\hat{y}_i$ is the predicted value for the i-th observation, $f_k(x_i)$ represents the prediction of the k-th tree for the i-th observation, and K is the number of trees.

XGBoost optimizes both the model complexity (number of trees, tree depth) and the regularization term to prevent overfitting. It also supports parallel and distributed computing for scalability.

Our methodology's culmination involves consolidating the insights gleaned from the predictive modeling process and leveraging the findings to inform future decision-making and strategies related to music curation and promotion. The robustness and reliability of the models pave the way for its practical application in real-world scenarios, enabling stakeholders in the music industry to make informed decisions regarding track selection, marketing strategies, and audience engagement initiatives. Additionally, our methodology lays the groundwork for ongoing refinement and optimization of predictive models to adapt to evolving trends and preferences within the dynamic landscape of the music industry.

IV. Data Set

We have built the dataset of Spotify Bengali Tracks(SBT) from scratch for this project. We wrote a script using Python and used Spotify Web API to fetch the track data from Spotify. Then we transformed our dataset into one containing categorical values using various thresholds for different attributes.

Our dataset provides comprehensive information about Spotify Bengali tracks encompassing a diverse collection across various genres. The dataset comprises multiple columns, each representing distinctive audio features associated with individual tracks.

SBT contains **4376** Bengali tracks from Spotify across diverse genres. The columns of the dataset include:

1) ID [string] – Unique IDs for each track.
2) Title [string] – The name of each track.
3) Artists [array of Artist object] – The artist/artists who performed in the track.
4) Popularity (target class) [string] –
   - Not Popular (0 - 10 of the raw dataset)
   - Kind of Popular (11 - 30 of the raw dataset)
   - Popular (31 - 50 of the raw dataset)
   - Very Popular (51 - 100 of the raw dataset)
5) Year [integer] - The Year when the track is released.
6) Danceability [Integer] –
   - No-dance (0 - 0.49 of the raw dataset)
   - Mid-dance (0.5 - 0.69 of the raw dataset)
   - High-dance (0.7 - 1.00 of the raw dataset)
7) Energy [string] –
   - Low Energy (0 - 0.50 of the raw dataset)
   - High Energy (0.51 - 1.00 of the raw dataset)
8) Key [string] –
   - En-harmonic Keys (values - 1, 3, 6, 8, 10 of the previous dataset)
   - Non-en-harmonic Keys values - 2, 4, 5, 7, 9, 11 of the previous dataset)
   - Unknown (value - -1 of the previous dataset)
9) Loudness [string] –
   - Loud (-11 or higher of the raw dataset)
   - Normal (-14 - -12 of the raw dataset)

- Quiet (-19 or less of raw dataset)

10) Speechiness [string] –
    - No Speech (0 - 0.33 of the raw dataset)
    - Mixed (0.34 - 0.66 of the raw dataset)
    - Speech (0.67 - 1.00 of the raw dataset)
11) Acousticness [string] –
    - Acoustic (0 - 0.50 of the raw dataset)
    - Non-Acoustic (0.51 - 1.00 of the raw dataset)
12) Instrumentalness [string] –
    - Vocal (0 - 0.50 of the raw dataset)
    - Instrumental (0.51 - 1.00 of the raw dataset)
13) Liveness [string] –
    - Recorded (0 - 0.79 of the raw dataset)
    - Live (0.8 - 1.00 of the raw dataset)
14) Valence [string] –
    - Negative (0 - 0.50 of the raw dataset)
    - Positive (0.51 - 1.00 of the raw dataset)
15) Tempo [string] –
    - Slow (0 - 110 of the raw dataset)
    - Medium (110 - 140 of the raw dataset)
    - Fast (141 or higher of the raw dataset)
16) Duration [string] –
    - Short (less than 3 minutes)
    - Medium (3 - 6 minutes)
    - Long (more than 6 minutes)
17) Time_Signature [string] –
    - Simple (0 - 4 of the raw dataset)
    - Complex (4 - 7 of the raw dataset)
    - Compound (7 or higher of the raw dataset)

V. Result and Analysis

In our investigation, we embarked on a comprehensive analysis leveraging a rich dataset encompassing more than 4300 Bengali songs sourced directly from Spotify. Our primary objective revolved around the exploration of the feasibility of predicting song popularity. To achieve this, we employed a diverse array of machine learning algorithms, ranging from Random Forest, Logistic Regression, k-nearest Neighbors (KNN), Support Vector Classifier (SVC), Gaussian Naive Bayes, and Decision Tree, to XGBoost.

One pivotal aspect of our analysis involved the thorough examination of confusion matrices generated by each employed algorithm. These matrices offered invaluable insights into the models' predictive performance by delineating the classification outcomes. Specifically, they allowed us to discern between true positives (correctly predicted popular songs), true negatives (correctly predicted non-popular songs), false positives (incorrectly predicted popular songs), and false negatives (incorrectly predicted non-popular songs). By scrutinizing these matrices, we gained a deeper understanding of the algorithms' efficacy in distinguishing between popular and non-popular songs, thereby illuminating their strengths and potential areas for improvement.

		Predicted	
		Positive (P)	Negative (N)
Actual	Positive (P)	TP	FN
	Negative (N)	FP	TN

TABLE I: Confusion Matrix

Here,
- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

And the performance measures,

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

The results of our applied algorithms are shown below in a tabulated form:

Algorithm	Accuracy	F1 Score	Precision	Recall
Logistic Regression	0.52	0.373	0.386	0.52
Random Forest	0.474	0.4118	0.396	0.474
K-Nearest Neighbors	0.48	0.447	0.432	0.482
Support Vector Classifier	0.529	0.377	0.488	0.529
Naive Bayes	0.0468	0.0428	0.550	0.0468
Decision Tree	0.48	0.425	0.4037	0.48
XGBoost	0.5028	0.416	0.4025	0.5028

TABLE II: Comparison of different algorithms

The performance of various machine learning algorithms on the dataset reveals different strengths and weaknesses. Support Vector Machine achieved the highest accuracy (0.529) but showed an imbalance in precision (0.488) and recall (0.529), indicating a potential preference for one class. XGBoost displayed balanced results across all metrics with an accuracy of 0.5028 and an F1 score of 0.416, making it a strong overall performer. Logistic Regression and K-nearest neighbors both had similar accuracies (0.52 and 0.48) and moderate F1 scores, suggesting they handle classes consistently. Random Forest had a slightly lower accuracy (0.474) but a reasonable F1 score (0.4118), indicating room for improvement. Decision Tree also performed moderately (accuracy of 0.48 and F1 score of 0.425). Naive Bayes, however, had very low performance across all metrics, suggesting it may not be suitable for this dataset. Overall, XGBoost and Support Vector Machine are promising options for further exploration and optimization.

VI. Conclusion

In this study, we aimed to predict song popularity by analyzing audio features of Bengali tracks sourced from Spotify. Utilizing the Spotify Web API, we curated a dataset of Bengali songs and processed it into categorical attributes to optimize results. Employing seven machine-learning methods, we assessed their performance on the dataset. SVM achieved the highest accuracy but showed an imbalance in precision and recall. XGBoost demonstrated balanced results, while Logistic Regression and KNN performed comparably. Random Forest and Decision Tree yielded moderate results, whereas Naive Bayes showed poor performance across all metrics, indicating unsuitability for the dataset.

In conclusion, XGBoost and SVM emerge as promising candidates for further exploration in predicting the popularity of Bengali songs on Spotify. Despite Bengali songs receiving comparatively lower listenership on Spotify, the dynamic nature of the music industry and growing interest in diverse cultural content suggest potential for increased traction in the future. Continued research and refinement in predictive models tailored to Bengali tracks could provide valuable insights for enhancing the visibility and commercial success of Bengali music on Spotify.

References

[1] Joshua S. Gulmatico, Julie Ann B. Susa, Mon Arjay F. Malbog, Aimee Acoba, Marte D. Nipas, and Jennalyn N. Mindoro. Spotipred: A machine learning approach prediction of spotify music popularity by audio features. In *2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T)*, pages 1–5, 2022.

[2] Brendan Joseph Suh. International music preferences: An analysis of the determinants of song popularity on spotify for the u.s., norway, taiwan, ecuador, and costa rica. 2019.

[3] Yap Kah Yee and Mafas Raheem. Predicting music popularity using spotify and youtube features. *Indian Journal of Science and Technology*, 15:1786–1799, 2022.

[4] Carlos Araujo, Marco Cristo, and Rafael Giusti. Predicting music popularity using music charts. pages 859–864, 12 2019.

[5] Jing Ren and Robert Kaufman. Understanding music track popularity in a social network. 06 2017.

[6] Rahman Autul, Dey D., Pritom Paul, and Raihan Ullah M. Rocking across borders : An analysis of the musical differences between bangladesh and west-bengal rock songs using spotify audio features. 04 2024.

[7] Patrick Helmholz, Michael Meyer, and Susanne Robra-Bissantz. Feel the moosic: Emotion-based music selection and recommendation. 06 2019.

[8] Yu-Yao Chang and Yao-Chung Lin. Music tempo (speed) classification. 01 2005.