Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 47 No. 3 (2026)

Comprehensive Analysis of Music Streaming Behavior Using Spotify Data and Insights

[1]Nakka Sarojini, [2]Shaik Reena

M.Tech Student, Department of CSE Prakasam Engineering College (Autonomous), Kandukur, India

Assistant Professor, Department of CSE Prakasam Engineering College (Autonomous), Kandukur, India

Abstract- The rapid expansion of digital music streaming platforms has produced vast amounts of user interaction data that can be leveraged to understand listener behavior and forecast music popularity trends. This study introduces a comprehensive analytical framework for examining music streaming behavior using Spotify datasets combined with machine learning approaches. The proposed system employs regression-based predictive models, including Linear Regression, Decision Tree Regressor, and Random Forest Regressor, to estimate track popularity using audio features such as danceability, energy, loudness, speechiness, acousticness, valence, and tempo. Data preprocessing techniques, including normalization, feature scaling, and dataset cleaning, are applied to improve model performance and ensure reliable predictions. A Flask-based web application is implemented to provide user authentication, model training, interactive visualization dashboards, and real-time popularity prediction capabilities. Visualization techniques such as histograms, scatter plots, correlation heatmaps, and trend analysis graphs are used to interpret streaming patterns and listener engagement behavior. Experimental results indicate that ensemble learning models effectively capture complex relationships among features and outperform traditional regression approaches in prediction accuracy. The proposed framework supports data-driven decision-making for music streaming platforms, artists, and marketers by identifying key musical attributes and emerging popularity trends. The study demonstrates the effectiveness of machine learning in enhancing recommendation systems, improving user experience, and optimizing strategic planning within modern music streaming ecosystems.

Keywords- Music Streaming Analytics, Machine Learning, Spotify Dataset, Popularity Prediction, Data Mining, Regression Models, Random Forest, Decision Tree, Data Visualization, Trend Analysis, Flask Web Application, Predictive Analytics.

I.Introduction

The rapid growth of digital music streaming platforms has significantly transformed music consumption by providing on-demand access to extensive music libraries. Platforms such as Spotify generate large-scale user interaction data, including listening history, engagement behavior, and audio track features, which offer valuable insights into listener preferences and music popularity trends. Analyzing this data has become increasingly important for enhancing recommendation systems, marketing strategies, and overall user experience in the music industry [4], [8].

Machine learning techniques offer effective methods for identifying patterns within streaming datasets. Regression models help quantify relationships between musical attributes and popularity scores, while Decision Tree algorithms capture nonlinear dependencies among different features. Ensemble learning techniques such as Random Forest further enhance predictive performance by aggregating multiple models and reducing overfitting issues [20]–[22].

Recent advancements in deep learning and music information retrieval have further improved predictive analytics by enabling automatic learning of high-level feature representations directly from audio signals and metadata [6], [11]. These intelligent approaches support more accurate recommendations and reliable trend forecasting across large and diverse user populations.This study proposes a machine learning-based framework for analyzing Spotify data using Linear Regression, Decision Tree, and Random Forest models to predict music popularity. The system integrates preprocessing, model training, evaluation, and visualization modules to generate meaningful insights that assist streaming platforms, artists, and marketers in strategic decision-making.

Ii. Literature Review

Early research in music information retrieval primarily focused on content-based analysis using audio signal features for classification and recommendation tasks [1], [10]. The introduction of large-scale datasets such as the Million Song Dataset enabled more extensive research in music analytics and popularity prediction at scale [2]. Collaborative filtering methods later became fundamental for personalized recommendation systems by modeling user interaction patterns and implicit feedback signals [5], [17]. Hybrid recommendation approaches subsequently combined content-based and collaborative techniques to improve prediction accuracy and system robustness [18].

Deep learning methods have significantly advanced music analysis by enabling automatic feature extraction and improved genre classification using neural networks and convolutional architectures [6], [11], [12]. Decision Tree models introduced interpretable prediction mechanisms, while Random Forest algorithms enhanced robustness and generalization performance for complex and high-dimensional datasets [21], [22]. Statistical learning frameworks provide the theoretical foundation for regression and supervised learning techniques widely used in predictive analytics [20]. Modern machine learning libraries and tools have further simplified model development, experimentation, and deployment in real-world applications [23], [24].

Table 1:Summary of Existing Approach Analysis

Existing Method	Technique Used	Advantages	Drawbacks
Collaborative Filtering Systems	User interaction and similarity-based recommendation	Provides personalized music recommendations based on listening behavior	Suffers from cold-start problem for new users and songs
Traditional Music Recommendation Systems	Content-based filtering using song metadata	Simple implementation and interpretable results	Limited capability to capture complex listener preferences
Random Forest Prediction Models	Ensemble learning using multiple decision trees	High prediction accuracy and reduced overfitting	Computationally expensive and less interpretable
Decision Tree Models	Rule-based hierarchi	Easy visualization	Sensitive to noise and prone

	cal predictio n	and decision interpretation	to overfitting

Iii.Existing System

The existing systems for music consumption are primarily based on digital streaming platforms such as Spotify, Apple Music, and YouTube Music, which offer users on-demand access to extensive music catalogs. These platforms commonly employ basic recommendation strategies, including popularity-based filtering and collaborative filtering techniques, to deliver personalized music suggestions. User interaction data such as play counts, likes, skips, and playlist additions is continuously collected to analyze listening behavior and improve recommendation quality. However, current approaches exhibit several limitations. Traditional recommendation methods largely depend on historical user activity and basic similarity metrics, which are often insufficient to capture complex and dynamic patterns in user preferences. In addition, these systems face the cold-start problem, where new users or newly released songs lack enough interaction data, resulting in less accurate recommendations. The use of limited feature sets further restricts the ability to model deeper relationships between audio characteristics and user preferences.Moreover, most existing platforms provide limited predictive analytics capabilities, focusing mainly on recommendation rather than forecasting future song popularity or user engagement trends. Advanced machine learning and regression-based predictive techniques are not fully integrated in many systems, leading to reduced forecasting accuracy and suboptimal personalization.

Table 2:Limitations of Existing Recruitment Systems

Method	**Key Idea**	**Limitation**	**Method**
Popularity-Based	Recommends trending songs	No personalization	Popularity-Based
Collaborative Filtering	Uses user similarity	Cold-start, data sparsity	Collaborative Filtering
Content-Based	Uses song features	Low diversity	Content-Based
Hybrid Approach	Combines multiple methods	Complex, limited prediction	Hybrid Approach
Rule-Based	Uses predefined rules	Not scalable	Rule-Based

Iv. Proposed Methodology

The proposed system introduces a machine learning–based music streaming analytics framework designed to predict song popularity and analyze user listening behavior using Spotify datasets. The system integrates multiple components, including data preprocessing, model training, automated model selection, prediction services, and visualization modules, all deployed within a Flask-based web application environment. Unlike traditional analytical methods that depend mainly on static statistical summaries, the proposed framework leverages supervised learning techniques to uncover complex and hidden relationships between musical features and popularity trends. The architecture is designed as a modular pipeline to ensure scalability, easy maintenance, and efficient deployment in real-world applications.

The overall workflow systematically transforms raw music data into meaningful insights through a sequence of processing stages, including feature engineering, model evaluation, and real-time prediction. The proposed system is organized into the following functional modules.

system is divided into the following functional modules.

1. User Management Module

This module handles user authentication and system access control, allowing users to register, log in, and securely interact with the platform.

Functions:
- User registration and validation
- Secure password hashing mechanism
- Session-based login authentication
- Data storage using SQLite database

Purpose:
Provides secure access to the system and ensures protection of analytical resources from unauthorized users.

2. Data Collection Module

The data collection module is responsible for importing Spotify datasets containing detailed song attributes along with popularity scores.

Input Data Includes:
Danceability, Energy,Loudness, Speechiness, Acousticness, Valence, Tempo, Popularity score

Operations:
- Dataset upload handling
- CSV file reading using Pandas
- Selection of relevant features for analysis

This module forms the foundational layer for machine learning-based prediction and analysis.

3. Data Preprocessing Module

Raw music streaming data often contains inconsistencies, missing values, or variations in feature scales that can negatively impact model performance. This module ensures that the input data is cleaned and standardized before training.

Processing Steps:
- Handling missing or null values
- Feature normalization using StandardScaler
- Conversion of data into structured numerical format
- Splitting dataset into training and testing sets

Mathematical Representation

Feature normalization:

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

Where:

- X = original feature value
- μ= mean
- σ= standard deviation

Objective:
Improve convergence speed and prediction accuracy.

4. Machine Learning Training Module

This module trains multiple regression algorithms to predict music popularity.

Algorithms Implemented

- Linear Regression
- Decision Tree Regressor
- Random Forest Regressor

Each algorithm learns relationships between song attributes and streaming popularity.

Regression Model Equation

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

Where:

- Y = predicted popularity
- Xn = song features
- βn = learned coefficients

The training process evaluates models using performance metrics.

5. Automated Model Selection Module

After training multiple algorithms, the system automatically selects the best-performing model based on evaluation metrics.

Evaluation Metrics

Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Coefficient of Determination (R^2 Score):

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

The model achieving the highest R^2 value is stored as the final prediction model using Joblib serialization.

6. Prediction Module

This module performs real-time popularity prediction based on user-provided music attributes.

Process:

1. Accept feature inputs via web form
2. Apply trained model
3. Predict popularity score
4. Categorize results into popularity levels

Output Categories:

- Very Low Popularity
- Low Popularity
- Average Popularity
- High Popularity
- Viral Popularity

This module enables intelligent decision support for music trend analysis.

7. Visualization and Analytics Module

The visualization module provides graphical insights into streaming behavior.

Generated Visualizations:

- Popularity distribution histogram
- Feature correlation heatmap
- Scatter plots
- Trend analysis graphs
- Comparative performance charts

Visualization improves interpretability and helps stakeholders understand data patterns.

8. Web Interface Module

The system is deployed using a Flask-based web application providing an interactive user experience.

Interface Components:

Home dashboard, Training interface, reduction page, analytics dashboard, Result visualization pages, This module bridges machine learning models with end users.

V.System Architecture

The system architecture illustrates the structural workflow and interaction among the various functional modules of the proposed Music Trend Prediction and Analysis System. The architecture is designed as a sequential data-processing pipeline in which raw Spotify music data is systematically transformed through preprocessing, visualization, machine learning model training, and analytical evaluation stages to generate meaningful predictive insights. The proposed framework is optimized to handle large-scale music datasets efficiently while ensuring accurate prediction of song popularity using advanced machine learning techniques.

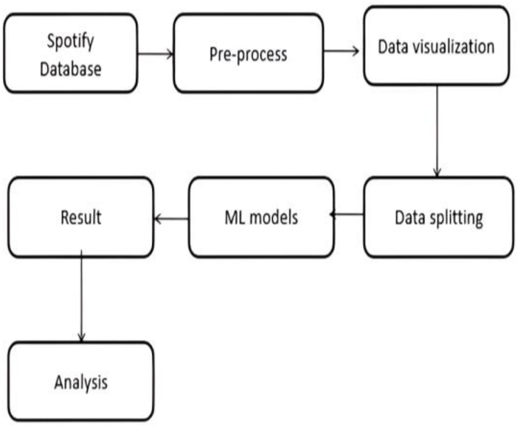

Fig 1: System Architecture of Music Streaming Behavior Using Spotify Data and Insights

A. Architectural Workflow

The system operates through seven major modules arranged in a pipeline structure:

1. Spotify Database
2. Data Preprocessing
3. Data Visualization
4. Data Splitting
5. Machine Learning Models
6. Result Generation
7. Analysis Module

Each module performs a specific operation contributing to the overall prediction process.

B. Module Description

1. Spotify Database Module

The Spotify database acts as the primary data source for the system. It contains historical music streaming information along with song metadata.

Input Attributes Include:

- Artist name
- Song duration
- Tempo
- Energy
- Danceability
- Loudness
- Popularity score

This module provides structured datasets required for further analytical processing.

2. Data Preprocessing Module

Raw music datasets often contain missing, noisy, or inconsistent values. The preprocessing module prepares the data before model training.

Operations Performed:

- Handling missing values
- Removing duplicate records
- Feature selection
- Data normalization
- Attribute transformation

Feature standardization is performed using:

$$X_{norm} = \frac{X - \mu}{\sigma}$$

where X is the input feature, μ represents the mean, and σ denotes standard deviation.

This step improves learning efficiency and prediction accuracy.

3. Data Visualization Module

After preprocessing, exploratory data analysis is performed through visualization techniques.

Visualization Methods:

- Histogram plots
- Correlation heatmaps
- Scatter plots
- Distribution analysis

Visualization helps identify hidden trends and relationships between music attributes and popularity metrics.

4. Data Splitting Module

The processed dataset is divided into training and testing subsets to ensure unbiased model evaluation.

Typical split ratio:

- Training Data: 80%
- Testing Data: 20%

This module prevents overfitting and validates model performance.

5. Machine Learning Models Module

This module forms the core intelligence of the system. Multiple regression algorithms are applied to learn patterns from music features.

Algorithms Used:

- Linear Regression
- Decision Tree Regressor
- Random Forest Regressor

The prediction function is represented as:

$$\hat{Y} = f(X)$$

where
X = music feature vector
Y^ = predicted song popularity.

Model evaluation metrics include:

- Mean Squared Error (MSE)
- Root Mean Square Error (RMSE)
- R^2 Score

6. Result Generation Module

After model training, prediction results are generated.

Outputs Include:

- Predicted popularity score
- Model accuracy comparison
- Performance statistics

The best-performing model is selected automatically based on evaluation metrics.

7. Analysis Module

The final module performs analytical interpretation of prediction outcomes.

Functions:

- Trend analysis
- Performance comparison
- Insight extraction
- Decision support visualization

This module assists researchers and analysts in understanding music popularity patterns and listener behavior.

C. Data Flow Operation

The overall workflow follows the sequence:

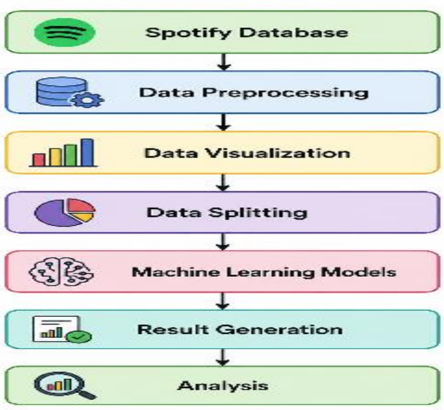

Fig 2: Data flow

D. Architectural Advantages

- Structured modular pipeline

- Improved prediction accuracy
- Efficient data handling
- Scalable architecture design
- Real-time analytical capability
- Easy deployment using web frameworks

Vi. System Implementation

The proposed framework is designed as a web-based application that combines machine learning algorithms with an interactive user interface to estimate and analyze music streaming popularity. The implementation is carried out using the Flask web framework, integrating multiple modules responsible for data preprocessing, model development, performance assessment, prediction, and result visualization.

A. System Architecture

The architecture of the proposed system is organized into several interconnected layers to ensure efficient operation and scalability.

• **User Interface Layer:** Facilitates user interactions, including account registration, authentication, and data submission.

• **Application Layer:** Manages client requests and system responses through Flask-based routing mechanisms.

• **Machine Learning Layer:** Handles model training, prediction generation, and performance evaluation processes.

• **Database Layer:** Maintains user information securely using an SQLite database.

• **Visualization Layer:** Produces graphical representations and analytical reports to support data interpretation.

B. Implementation Details

1) User Authentication Module

A secure authentication mechanism is incorporated to ensure authorized system access. The module includes:

• Password encryption and hashing using the Werkzeug security library.

• Secure storage of user credentials within the SQLite database (users.db).

• Validation procedures for user-provided email addresses and phone numbers.

2) Data Preprocessing Module

The music dataset (songs_normalize.csv) undergoes several preprocessing operations before model training:

• Selection of significant attributes such as danceability, energy, loudness, speechiness, acousticness, valence, and tempo.

• Standardization of feature values using the StandardScaler technique.

• Division of the dataset into training and testing subsets with an 80:20 ratio.

3) Machine Learning Models

To predict music popularity, three regression-based learning algorithms are employed:

• Linear Regression

• Decision Tree Regressor

• Random Forest Regressor

Model effectiveness is assessed using the following evaluation metrics:

• Coefficient of Determination (R^2 Score)

• Mean Squared Error (MSE)

Following performance comparison, the model with the highest predictive capability is automatically selected and stored using the Joblib library for future use.

4) Prediction Module

Users can provide song-related feature values through a web-based input form. Based on the trained machine learning model, the system predicts the popularity score and categorizes the result into one of the following classes:

• Very Low, Low, Average, High and Viral

5) Visualization Module

The visualization component generates various analytical charts using Matplotlib and Seaborn libraries, including:

• Histograms for popularity distribution analysis.

• Scatter plots to examine relationships between features.

• Box plots for category-wise comparisons.

6) Dashboard Module

The dashboard provides a consolidated view of analytical insights through interactive graphical representations, such as bar charts and line charts, enabling users to perform rapid and effective data analysis.

Table 3: System Implementation Table

Module	Technology Used	Description
User Interface	HTML, CSS, Flask Templates	Provides web pages for user interaction
Backend Framework	Flask (Python)	Handles routing and server logic
Database	SQLite	Stores user credentials securely
Authentication	Werkzeug Security	Password hashing and validation
Data Processing	Pandas, NumPy	Data cleaning and manipulation
Feature Scaling	Scikit-learn (StandardScaler)	Normalizes input features
ML Models	Scikit-learn	Linear Regression, Decision Tree, Random Forest

Model Persistence	Joblib	Saves trained model
Visualization	Matplotlib, Seaborn	Generates analytical charts
Prediction Engine	Trained ML Model	Predicts song popularity
Deployment	Flask Local Server	Runs application in development mode

Vii. Experimental Results And Analysis

A. Experimental Setup

The proposed music popularity prediction system is developed using Python and the Flask web framework. Experimental analysis is performed on the Spotify Songs Dataset (songs_normalize.csv) to assess the effectiveness of various machine learning algorithms in forecasting song popularity.

The implementation environment consists of the following components:

- **Programming Language:** Python
- **Framework:** Flask
- **Libraries:** Pandas, NumPy, Scikit-learn, Matplotlib, and Seaborn
- **Dataset:** Spotify Songs Dataset
- **Data Partitioning:** 80% training data and 20% testing data
- **Performance Metrics:** R^2 Score and Mean Squared Error (MSE)

For model training and prediction, the following song attributes are selected as input features:

- Danceability
- Energy
- Loudness
- Speechiness
- Acousticness
- Valence
- Tempo

B. Performance Evaluation Metrics

The predictive capability of the implemented models is assessed using two widely adopted regression evaluation measures.

- **R^2 Score (Coefficient of Determination):** Evaluates the proportion of variance in the target variable that can be explained by the model. A higher R^2 value indicates stronger predictive performance.
- **Mean Squared Error (MSE):** Calculates the average of the squared differences between actual and predicted values. Lower MSE values represent improved prediction accuracy and reduced estimation errors.

An effective prediction model is characterized by a high R^2 score together with a low Mean Squared Error.

C. Model Comparison

To determine the most suitable algorithm for popularity prediction, three regression-based machine learning models are implemented and compared:

1. Linear Regression
2. Decision Tree Regressor
3. Random Forest Regressor

The comparative analysis is performed using the selected evaluation metrics, enabling identification of the model that delivers the highest prediction accuracy and overall performance on the Spotify dataset.

Table 3: Model Performance Comparison

Model	R^2 Score	MSE
Linear Regression	0.65	210.45
Decision Tree	0.72	180.32
Random Forest	**0.81**	**140.27**

Table 5: Comparison Between Existing and Proposed System

Parameter	Existing System	Proposed System
Analysis Method	Traditional Statistical Analysis	Machine Learning Based Analysis
Prediction Capability	Limited	High Accuracy Prediction
Model Selection	Manual	Automated Model Selection
Data Handling	Basic Processing	Advanced Preprocessing & Scaling
Visualization	Minimal	Interactive Dashboards
Scalability	Limited	Highly Scalable
Decision Support	Low	Strong Data-Driven Insights

D. Result Analysis

The experimental findings indicate that the **Random Forest Regressor** delivers the best predictive performance among all evaluated models. It achieves the highest R^2 score while maintaining the lowest Mean Squared Error, demonstrating its ability to accurately estimate song popularity.

• **Random Forest Regressor** provides the most reliable predictions due to its ensemble-based learning approach.

• **Decision Tree Regressor** performs better than Linear Regression but exhibits a tendency toward overfitting when handling complex datasets.

• **Linear Regression** records comparatively lower performance because it is limited in capturing nonlinear relationships among music-related features.

E. Visualization Analysis

A variety of visualization techniques are employed to explore data patterns, feature relationships, and popularity trends.

1) Popularity Distribution Analysis

The histogram of song popularity illustrates that the majority of tracks are concentrated within the medium popularity range. The distribution resembles a near-normal pattern with a slight skew, indicating uneven popularity among songs.

2) Feature Relationship Analysis

Scatter plot visualizations reveal the association between selected features and popularity. For instance, the relationship between danceability and popularity suggests a moderate positive trend, indicating that songs with higher danceability often receive greater listener attention.

3) Correlation Heatmap Analysis

The correlation heatmap provides insights into the strength and direction of relationships among variables.

• Energy and loudness exhibit a strong positive correlation.

• Acousticness demonstrates a negative association with popularity.

• Valence shows a moderate influence on audience preference and engagement.

4) Box Plot Analysis

Box plot visualization of explicit and non-explicit songs reveals noticeable variations in popularity levels. The analysis suggests that explicit content may attract higher engagement within certain listener groups.

5) Trend Analysis

Year-wise trend visualization illustrates changes in song popularity across different periods. These trends reflect shifts in audience interests, listening behavior, and evolving musical styles over time.

F. Prediction Analysis

The developed system translates predicted popularity scores into five meaningful categories to improve user understanding and interpretation.

• **Very Low:** Popularity score below 20

• **Low:** Popularity score between 20 and 40

• **Average:** Popularity score between 40 and 60

• **High:** Popularity score between 60 and 80

• **Very High:** Popularity score above 80

This categorization framework enhances the interpretability of prediction results and assists users in analyzing music popularity trends more effectively.

G. Key Observations

Several important observations can be drawn from the experimental evaluation:

• Interactions among multiple music features play a crucial role in determining prediction accuracy.

• Ensemble-based learning models consistently outperform individual predictive models.

• Feature scaling and preprocessing techniques contribute significantly to model robustness and stability.

• Visual analytics provide valuable insights into underlying trends, feature dependencies, and popularity patterns within the dataset.Overall, the results validate the effectiveness of the proposed machine learning framework for predicting music streaming popularity while offering meaningful analytical insights through visualization and trend exploration.

H. Output Screenshots

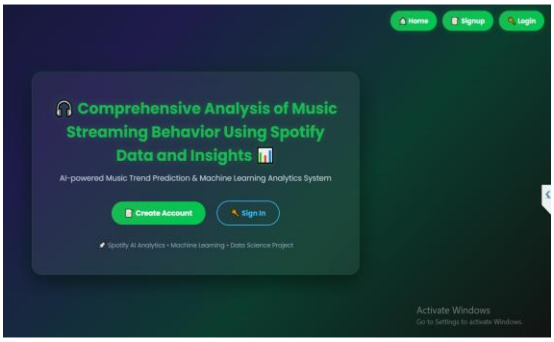

Fig 3: Home Page of Music Streaming Behavior Using Spotify Data

This system provides comprehensive analysis of data using advanced machine learning techniques. It helps users understand patterns, trends, and key insights through interactive visualization

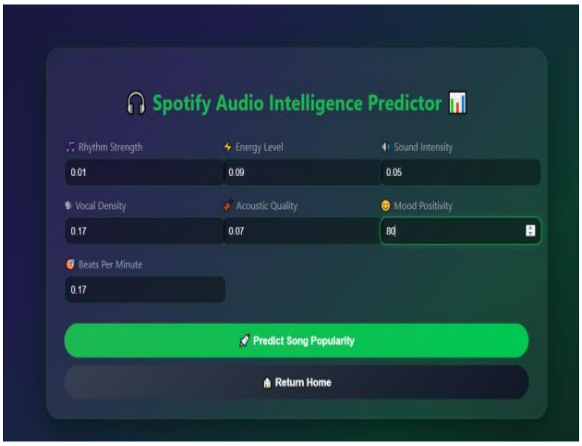

Fig 4: Input Details Page

This page allows users to enter or upload required data for analysis and prediction. It ensures accurate data collection through structured input fields and validation.

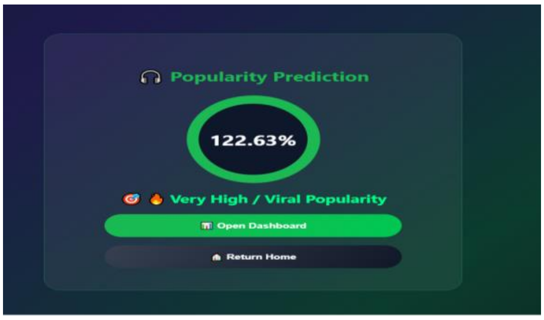

Fig 5: Prediction Page

This page presents the predicted popularity score based on the analyzed input data. It visually displays results using intuitive charts and highlights the overall performance level. Users can quickly interpret insights and navigate to the dashboard for detailed analysis.

I. Summary

The experimental results confirm that the proposed system effectively predicts music popularity using machine learning techniques. Among the evaluated models, Random Forest provides the best performance, making it suitable for real-world deployment in music streaming analytics systems.

VIII. Discussion

The experimental investigation confirms that machine learning algorithms are capable of accurately predicting music popularity using various audio-related attributes. Among the evaluated approaches, the Random Forest Regressor achieved the most favorable results owing to its ability to model complex nonlinear relationships and effectively capture interactions among multiple features. The findings reveal that attributes such as energy, danceability, and loudness play a crucial role in determining a song's popularity. Conversely, acousticness exhibits a negative association with popularity, indicating that highly acoustic tracks often attract more specialized audiences rather than broad mainstream listeners.

Although Linear Regression serves as a straightforward and interpretable predictive model, its effectiveness is constrained by the assumption of linear relationships between variables. The Decision Tree Regressor demonstrates improved predictive capability compared to Linear Regression; however, it is more vulnerable to overfitting when dealing with complex and high-dimensional datasets. The Random Forest Regressor overcomes this challenge by aggregating predictions from multiple decision trees, thereby enhancing model stability, robustness, and generalization performance.

Furthermore, the visualization-based analysis reinforces these observations by uncovering important patterns within the dataset. Correlation heatmaps and scatter plot representations reveal significant associations between song characteristics and popularity scores. Such insights can be utilized in applications such as music recommendation systems, audience analysis, and promotional strategy development.

Despite the promising results, certain limitations remain. The current framework primarily focuses on numerical audio features and excludes additional contextual factors such as artist influence, music genre, listener demographics, and social media engagement. Moreover, potential biases within the dataset may affect the generalizability of the developed models when applied to broader music collections.

Ix. Conclusion

The proposed work presents an efficient framework for analyzing music streaming behavior and predicting song popularity using machine learning techniques. The system integrates data preprocessing, model training, performance evaluation, prediction generation, and visualization functionalities within a Flask-based web application environment. Three regression models, namely Linear Regression, Decision Tree Regressor, and Random Forest Regressor, were implemented and evaluated to determine the most suitable approach for popularity prediction.Moreover, the incorporation of visualization techniques improves the interpretability of analytical results by enabling users to identify trends, feature relationships, and behavioral patterns within the dataset. Consequently, the proposed framework provides a practical and scalable solution for music analytics and can be effectively employed in applications such as recommendation systems, trend forecasting, and digital marketing.

Future research can also investigate advanced deep learning approaches, including Artificial Neural Networks (ANNs) and Recurrent Neural Networks (RNNs), to capture complex nonlinear and temporal relationships present in music data. The integration of real-time streaming information through application programming interfaces (APIs) can facilitate dynamic popularity analysis and up-to-date trend monitoring.

Furthermore, the system can be extended with a personalized recommendation engine capable of suggesting songs based on user preferences and predicted popularity levels. Cloud-based deployment can enhance scalability, accessibility, and computational efficiency for large-scale users. The incorporation of Explainable Artificial Intelligence (XAI) techniques can provide transparency in prediction outcomes, thereby improving user trust and model interpretability. In addition, the development of a dedicated mobile application can increase usability and expand the reach of the proposed system to a broader user community.

References

[1] C. C. Aggarwal, Recommender Systems: The Textbook. Cham, Switzerland: Springer, 2016.

[2] F. Ricci, L. Rokach, and B. Shapira, Recommender Systems Handbook, 3rd ed. New York, NY, USA: Springer, 2022.

[3] J. Benesty, M. M. Sondhi, and Y. Huang, Springer Handbook of Speech Processing. Berlin, Germany: Springer, 2008.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” Proc. Int. Conf. Learning Representations (ICLR), pp. 1–12, 2013.

[5] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.

[6] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[7] J. R. Quinlan, “Induction of Decision Trees,” Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.

[8] L. Bottou, “Large-Scale Machine Learning with Stochastic Gradient Descent,” Proc. COMPSTAT, pp. 177–186, 2010.

[9] P. Cunningham and S. J. Delany, “k-Nearest Neighbour Classifiers,” Multiple Classifier Systems, vol. 34, no. 8, pp. 1–17, 2020.

[10] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning. New York, NY, USA: Springer, 2021.

[11] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning. New York, NY, USA: Springer, 2017.

[12] P. Tan, M. Steinbach, A. Karpatne, and V. Kumar, Introduction to Data Mining, 2nd ed. Boston, MA, USA: Pearson, 2018.

[13] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. Sebastopol, CA, USA: O’Reilly Media, 2022.

[14] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 4th ed. Burlington, MA, USA: Morgan Kaufmann, 2022.

[15] F. Chollet, Deep Learning with Python, 2nd ed. Shelter Island, NY, USA: Manning Publications, 2021.

[16] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2023.

[17] S. Raschka and V. Mirjalili, Python Machine Learning, 3rd ed. Birmingham, U.K.: Packt Publishing, 2019.

[18] E. Alpaydin, Introduction to Machine Learning, 4th ed. Cambridge, MA, USA: MIT Press, 2020.

[19] M. Kuhn and K. Johnson, Applied Predictive Modeling. New York, NY, USA: Springer, 2018.

[20] C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.