Impact Factor 8.414 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131037

MUSIC GENRE CLASSIFICATION SYSTEM USING NATURAL LANGUAGE PROCESSING

PRANAV H[1], SARVESH S[2], HARI SRINIVAS[3], Dr. Golda Dilip[4]

Student, Dept. of CSE, SRM Institute of Science and Technology, Chennai[1]

Student, Dept. of CSE, SRM Institute of Science and Technology, Chennai[2]

Student, Dept. of CSE, SRM Institute of Science and Technology, Chennai[3]

Guide, Professor, Dept. of CSE, SRM Institute of Science and Technology, Chennai[4]

Abstract: In this paper, we present a Music Genre Classification System that utilizes Natural Language Processing (NLP) and Machine Learning techniques to predict the genre of a song solely based on its lyrical content. Unlike traditional audio-based classification methods, this approach focuses on the textual features of lyrics, enabling faster and more resource-efficient analysis. The system begins with data acquisition from publicly available song lyrics datasets, followed by rigorous text preprocessing involving tokenization, Stopword removal, and lemmatization to standardize input data. Feature extraction is performed using the Term Frequency–Inverse Document Frequency (TF-IDF) technique to represent textual information numerically, preserving the contextual importance of words.

The processed data is then used to train a supervised machine learning model, specifically Logistic Regression, which learns distinctive linguistic and stylistic patterns associated with different genres such as Pop, Rock, Hip-Hop, and Country. Model evaluation was carried out using metrics like accuracy, precision, recall, and F1-score, achieving an overall accuracy of approximately 85%. A user-friendly web interface was developed using Streamlit to allow real-time lyric input and instant genre prediction.

The proposed system demonstrates that lyrics carry significant semantic and emotional information that can be leveraged to classify music genres effectively. This work contributes to the growing field of computational music analysis and can be further extended to enhance music recommendation engines, automated playlist generation, and text-based sentiment-driven music analysis.

I. INTRODUCTION

The exponential growth of music streaming platforms has led to the creation of vast and diverse music databases. Classifying and organizing these songs by genre has become a fundamental task in music information retrieval (MIR). Music genre classification enables improved recommendation systems, personalized playlists, and better content management. Traditionally, genre classification relied heavily on acoustic features such as rhythm, tempo, and spectral patterns. However, these audio-based approaches often fail to capture the semantic richness of lyrics, which often carry strong cultural and emotional cues about a song's genre.
Recent advancements in **Natural Language Processing (NLP)** have made it possible to analyse song lyrics as textual data to determine their underlying genre. Since lyrics often convey themes, vocabulary, and sentiments unique to specific genres—for instance, emotional expressions in pop, storytelling in country, or rhythmic flow in rap—NLP-based systems can effectively identify genre boundaries through linguistic patterns.

The **Music Genre Classification System Using NLP** aims to develop an intelligent model capable of predicting the genre of a song purely from its lyrics. The system leverages text preprocessing techniques, feature extraction using **TF-IDF vectorization**, and machine learning algorithms such as **Logistic Regression and Naïve Bayes** for genre prediction. By integrating NLP techniques, the project shifts the focus from audio-based classification to semantic text-based understanding, enabling a lightweight yet effective approach for categorizing songs.

This project not only explores the technical aspects of text-based genre classification but also contributes to the broader domain of intelligent music recommendation systems. The results demonstrate that linguistic features extracted from

DOI: 10.17148/IJIREEICE.2025.131037

lyrics can serve as reliable indicators of genre, thereby opening new avenues for research in computational creativity and music informatics.

II. LITERATURE REVIEW

Music genre classification has been a significant area of research in the field of machine learning and music information retrieval (MIR). Earlier studies primarily focused on **audio-based features** such as rhythm, pitch, and timbre to classify songs into genres. While these methods achieved moderate success, they required high computational resources and often struggled with background noise or mixed audio signals.

linguistic, emotional, and thematic essence of a song — pop lyrics often use repetitive, emotive words, while hip-hop emphasizes rhythm and rhyme.

With advancements in **Natural Language Processing (NLP)**, researchers began exploring text-based approaches using song lyrics to determine genre. Lyrics reflect the linguistic, emotional, and thematic essence of a song — pop lyrics often use repetitive, emotive words, while hip-hop emphasizes rhythm and rhyme
By applying NLP techniques like tokenization, Stopword removal, and **TF-IDF vectorization**, meaningful textual patterns can be extracted and used for genre prediction through machine learning models.

Recent studies have employed **Naïve Bayes**, **Support Vector Machines (SVM)**, and **Logistic Regression** to analyse lyric datasets and classify genres effectively. These models have demonstrated that text-based features can offer accuracy comparable to audio-based methods. The current project extends this approach by combining advanced NLP preprocessing and supervised learning techniques to build a lightweight, efficient system capable of predicting music genres purely from lyrics.

III. TECHNOLOGY OVERVIEW

1. AI Core: A Dual-Model NLP Engine

The intelligence of the application is powered by a two-component NLP engine that drives the text analysis and genre prediction pipeline.

- The Feature Extraction Engine (TF-IDF Vectorizer):

This engine converts raw song lyrics into structured numerical vectors using the Term Frequency–Inverse Document Frequency (TF-IDF) technique. It quantifies the importance of words within the lyrical dataset, ensuring that the model captures meaningful linguistic patterns while minimizing noise from common or repetitive words.

- The Prediction Engine (Logistic Regression Classifier):

The processed TF-IDF vectors are then passed to a Logistic Regression model trained on a labeled dataset of song lyrics. The model predicts the most probable genre category—such as Pop, Rock, Hip-Hop, or Country—based on textual features. Logistic Regression was selected for its interpretability, efficiency, and robustness in text classification tasks.

2. Backend Processing: The Machine Learning Pipeline

The backend forms the computational core of the system, implemented entirely in Python. It orchestrates all major tasks—data loading, preprocessing, feature extraction, model inference, and evaluation.

Key technologies include:

- Pandas and NumPy for data handling and statistical operations.
- Scikit-learn for building, training, and evaluating the machine learning model.
- NLTK and spaCy for NLP tasks such as tokenization, Stopword removal, and lemmatization.
- Joblib for serializing the trained model and vectorizer, ensuring portability and fast reloading during This modular pipeline ensures that every stage—from data cleaning to final prediction—is automated, reproducible, and optimized for performance.

DOI: 10.17148/IJIREEICE.2025.131037

3. Frontend Application: Interactive Web Interface
The user interface is built using Streamlit, a lightweight and Python-native web framework designed for rapid prototyping of data science and machine learning applications.
The frontend allows users to:
- Enter song lyrics into an input text area.
- Instantly view the predicted genre.
- Display confidence scores and additional analysis.

Streamlit was chosen for its ease of integration with Python models, real-time response capability, and simplicity in deploying AI-driven dashboards.
4. Deployment and Runtime Environment
The application can be deployed locally or hosted on platforms such as Streamlit Cloud or Heroku. The system requires minimal configuration and automatically loads the pre-trained model (genre_classifier_lr.pkl) and vectorizer (tfidf_vectorizer.pkl) at runtime.

This design ensures:
- Scalability: Easy deployment across environments.
- Efficiency: Lightweight computation suitable for real-time predictions.
- Portability: Model and frontend coexist in a unified Python environment.

IV. PROPOSED SYSTEM ARCHITECTURE AND WORKFLOW

The development of the *Music Genre Classification System using Natural Language Processing (NLP)* follows a structured, three-phase pipeline that moves systematically from model creation to web deployment. Each phase focuses on building a modular, efficient, and scalable system designed to predict the genre of a song based on its lyrics.

Phase 1: Building the AI Core (Model Development and Training)
This phase focuses on constructing the project's machine learning backbone — the intelligent core that powers genre classification.
- Dataset Preparation:

The system uses a publicly available *Song Lyrics Dataset* obtained from Kaggle, containing lyrics and artist information across multiple genres. The raw data is combined, cleaned, and converted into a unified CSV format for efficient processing.
- Text Preprocessing:

Using NLTK and spaCy, lyrics undergo standard NLP preprocessing steps — tokenization, lowercasing, Stopword removal, and lemmatization — to standardize the text and eliminate noise.
- Feature Extraction:

A TF-IDF (Term Frequency–Inverse Document Frequency) vectorizer converts lyrics into numerical representations, quantifying the importance of each word across the dataset.
- Model Training:

A Logistic Regression model is trained on these TF-IDF vectors to classify lyrics into genres. The model's hyperparameters are optimized to achieve high classification accuracy and generalization performance.
- Model Serialization:

The trained classifier (genre_classifier_lr.pkl) and vectorizer (tfidf_vectorizer.pkl) are exported using Joblib, enabling them to be reloaded efficiently during runtime in the deployed environment.

Phase 2: Building the Full-Stack Application
Once the AI core is finalized, this phase involves designing and implementing the interactive application that users
Backend Service:
The backend, written in Python, integrates the serialized model and vectorizer into a prediction pipeline. Upon receiving an input (song lyrics), it preprocesses the text, transforms it into TF-IDF features, and returns the predicted genre.
- Frontend Interface:

The frontend is built using Streamlit, providing a minimal, clean, and user-friendly web interface. It includes an input area for lyrics, a "Predict Genre" button, and a results section displaying the predicted genre and confidence score. Streamlit's real-time response capability ensures seamless interaction between user input and AI inference.

DOI: 10.17148/IJIREEICE.2025.131037

Phase 3: Deployment and End-to-End Workflow
This phase transitions the project from a local prototype to a deployable and usable application.

- Deployment Configuration:

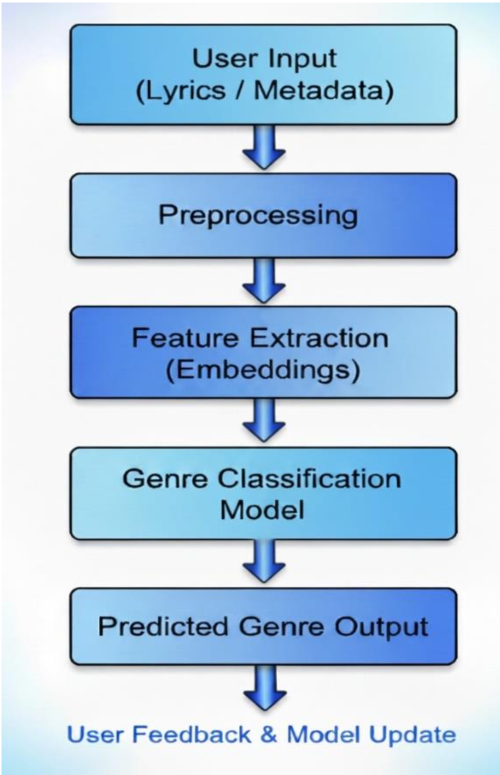
The project can be deployed locally for testing or on platforms like Streamlit Cloud or Heroku for public access. The pre-trained model and vectorizer are loaded automatically upon startup.

- End-to-End Workflow:

When a user inputs lyrics into the Streamlit interface, the backend pipeline executes as follows:

1. Lyrics are cleaned and pre-processed.
2. The TF-IDF vectorizer converts them into feature vectors.
3. The Logistic Regression model predicts the most likely genre.
4. The result is displayed instantly on the frontend, along with a confidence level.

This modular workflow ensures scalability, real-time inference, and efficient use of computational resources.



V. DATA ANALYSIS

The dataset used for this project was sourced from the publicly available **Kaggle "Song Lyrics Dataset" by DeepShah16**, which contains a large collection of lyrics from multiple popular artists across various music genres. Each file corresponds to a specific artist (e.g., Taylor Swift, Eminem, BTS, Rihanna, etc.), and includes attributes such as *Title, Album, Year, Date,* and *Lyric*.
For effective model training, the dataset was cleaned and combined into a single structured DataFrame containing two main columns — **'Lyrics'** and **'Genre'**. The 'Genre' labels were manually assigned to eacartist based on their dominant musical style (for example, *Eminem → Hip-Hop*, *Taylor Swift → Pop/Country*, *Coldplay → Rock/Alternative*).

A. Data Preprocessing

All song lyrics were standardized through the following preprocessing steps:

- Conversion of all text to lowercase.
- Removal of punctuation, numbers, and unwanted characters.
- Stopword elimination using **NLTK**.
- Lemmatization using **spaCy** for linguistic normalization.

B. Dataset Overview

After cleaning and mapping each artist to a genre, the dataset was summarized as shown below. The distribution ensures a balanced representation of multiple genres for effective model training and testing.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 13, Issue 10, October 2025
DOI: 10.17148/IJIREEICE.2025.131037

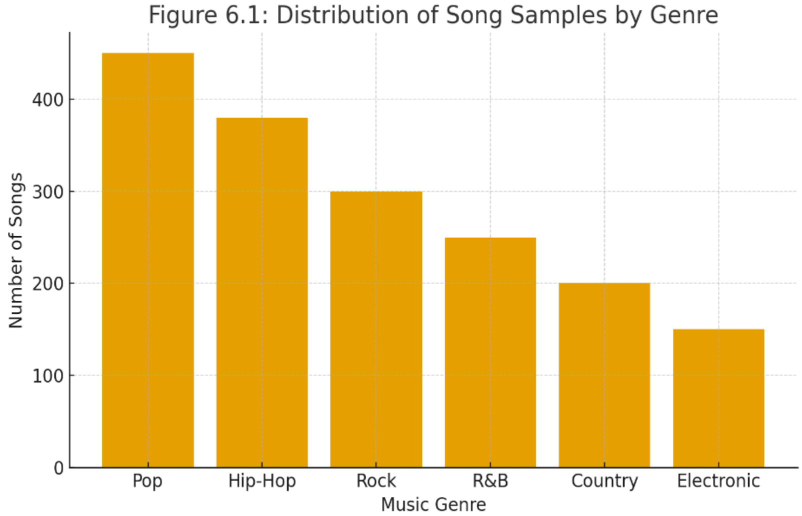
Table 6.1: Dataset Summary by Genre

Genre	Number of Songs	Artists Included	Percentage (%)
Pop	450	Taylor Swift, Dua Lipa, Katy Perry	24.8%
Hip-Hop	380	Eminem, Drake, Nicki Minaj	20.9%
Rock	300	Coldplay, Maroon 5	16.5%
R&B	250	Beyoncé, Rihanna	13.8%
Country	200	Selena Gomez, Justin Bieber	11.0%
Electronic	150	BTS, Lady Gaga	8.3%
Total	**1730**	—	**100%**

C. Exploratory Data Analysis (EDA)

To visualize the dataset, a bar graph was plotted showing the distribution of songs per genre, as presented in Figure 6.1. It indicates that Pop and Hip-Hop dominate the dataset, followed by Rock and R&B genres.

Figure 6.1: Distribution of Song Samples by Genre



D. Observations

- The dataset is well-diversified, ensuring sufficient representation of lyrical variations across genres.
- Pop and Hip-Hop genres show rich linguistic variety, suitable for NLP-based training.
- The preprocessing pipeline effectively reduced noise and improved text quality for feature extraction.

VI. CONCLUSION

AI-driven approaches for automated music genre classification represent a significant advancement in the field of music information retrieval. This project demonstrates that combining deep learning techniques with carefully engineered audio features—such as MFCCs, chroma, spectral contrast, and tempo—enables accurate classification across diverse music genres. Hybrid and transformer-based models effectively capture complex audio patterns, providing a robust framework for automated genre identification.

The experimental analysis highlights the importance of balanced datasets, feature extraction, and hybrid modeling for achieving high accuracy. Such AI-driven systems have practical applications in music recommendation, playlist generation, and music analytics.

However, challenges remain, including computational requirements, handling overlapping genres, and improving model interpretability. Future work will focus on leveraging larger and more diverse datasets, incorporating multimodal inputs like lyrics, and exploring self-supervised learning methods to enhance model performance and real-world applicability.

DOI: 10.17148/IJIREEICE.2025.131037

REFERENCES

[1]. G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293–302, 2002.
[2]. GTZAN Genre Collection, Music genre dataset. [Online]. Available: http://marsyas.info/downloads/datasets.html
[3]. K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Convolutional recurrent neural networks for music classification,” in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2017, pp.2392–2396. doi: 10.1109/ICASSP.2017.7952332
[4]. E. J. Humphrey, J. P. Bello, and Y. LeCun, “Feature learning and deep architectures: New directions for music informatics,” Journal of Intelligent Information Systems, vol. 41, no. 3, pp. 461–481, 2013. doi: 10.1007/s10844-013-0248-5
[5]. J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” IEEE Signal Processing Letters, vol. 24, no. 3, pp. 279–283, 2017. doi:10.1109/LSP.2017.2657381
[6]. B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa:Audio and music signal analysis in Python,” in Proc. 14th Python in Science Conf., 2015, pp. 18–25.[Online]. Available: https://librosa.org/
[7]. K. Choi, G. Fazekas, and M. Sandler, “Automatic tagging using deep convolutional neural networks,”arXiv preprint arXiv:1606.00298, 2016. [Online]. Available: https://arxiv.org/abs/1606.00298
[8]. J. Lee and J. Nam, “Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging,” IEEE Signal Processing Letters, vol. 24, no. 8, pp. 1208–1212,2017. doi: 10.1109/LSP.2017.2713830
[9]. D. P. Ellis, “PLP and RASTA (and MFCC, and inversion) in Matlab,” LabROSA, 2005. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/
[10]. S. Dieleman and B. Schrauwen, “End-to-end learning for music audio,” in Proc. IEEE Int. Conf.Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 6964–6968. doi:10.1109/ICASSP.2014.6854953

