Music Genre Classification Using Audio Data

Sowmya Natarajan[0000-0002-9888-6078], Shivansh Saxena[0009-0007-0937-5158],

Ishan Bhardwaj[0009-0005-6186-4069], Hussain Hakeem[0009-0000-3743-2550]

Dept of Electronics & Communication Engineering S.R.M. Institute of Science & Technology Chennai, India.
sowmyan1@srmist.edu.in, ss0885@srmist.edu.in
ib8884@srmist.edu.in, hh1834@srmist.edu.in

Abstract. Music genre classification involves automatically categorizing music tracks into specific genres based on audio features. Key techniques include feature extraction, focusing on attributes like rhythm, timbre, and pitch. Algorithms such as K-Nearest Neighbors (K-NN), Support Vector Machines (SVM), and Random Forests are commonly used to perform the classification task. K-NN relies on similarity measures between tracks, SVM separates genres by finding optimal hyperplanes, and Random Forests combine multiple decision trees for robust classification. These methods enhance music recommendation and organization systems by enabling efficient genre identification. We achieved highest accuracy of 87% with random forest, 71% with svm and 27% with KNN

Keywords: Music genre classification, Support Vector Machines (SVM), Random Forests

1 Introduction

Music genre classification has gained significant attention in recent years due to its applications in music recommendation systems and automated content organization. Support Vector Machines (SVM) have been widely used for this task, demonstrating the ability to classify music genres by identifying optimal hyperplanes that separate audio data into distinct categories based on features such as timbre and rhythm [1].

Additionally, the advent of deep learning, particularly Convolutional Neural Networks (CNNs), has further improved classification accuracy by automatically extracting complex patterns from raw audio data [2]. These advancements have made genre classification more accurate and efficient, paving the way for more sophisticated music retrieval systems. K-Nearest Neighbors (KNN) is another popular algorithm for music genre classification, leveraging proximity-based methods to classify tracks by comparing them to their nearest neighbors in the feature space. When combined with Principal Component Analysis (PCA) for dimensionality reduction, KNN has demonstrated significant performance improvements in classification tasks, offering a simple yet effective approach to identifying music genres [3]. Earlier studies also explored feature-based methods, such as the work by Tzanetakis and Cook, which focused on extracting key audio features like timbre, rhythm, and pitch to classify musical genres effectively

Research Paper
DOI: https://doi.org/10.46793/BISEC25.466N
Part of ISBN: 978-86-89755-40-4

[4]. These feature extraction techniques have laid the groundwork for modern classification models by emphasizing the importance of audio signal processing.

Despite the advancements in machine learning and deep learning for music genre classification, feature extraction remains a critical aspect of improving model accuracy. Novel feature extraction methods, such as those proposed by Hu and Li, have introduced more effective ways to capture the intricate characteristics of audio signals, leading to more accurate genre predictions [5]. These innovative techniques focus on refining the representation of audio data, allowing classifiers to better distinguish between genres. As the field continues to evolve, further enhancements in feature extraction and processing will be crucial for developing even more robust and scalable classification models.

2 Literature survey

The paper "A Comparative Study of Machine Learning Techniques for Music Genre Classification"[6] evaluates various machine learning models, including SVM, k-NN, ANN, Decision Trees, and Random Forests, to classify music genres. Using the GTZAN dataset, the study analyzes performance based on accuracy, comparing different algorithms and feature extraction methods like Mel Frequency Cepstral Coefficients (MFCC). The findings highlight the strengths and weaknesses of each model, emphasizing that ensemble methods like Random Forests and deep learning approaches tend to outperform traditional algorithms in music genre classification tasks. The paper "Deep Learning for Music Genre Classification: A Survey"[7] published in IEEE Access (2018) provides a detailed review of deep learning techniques used for music genre classification. The authors explore various deep learning models, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and hybrid architectures, discussing their effectiveness in extracting complex patterns from audio data.

The paper also highlights the advantages of deep learning over traditional machine learning methods, such as automatic feature extraction and improved accuracy. Additionally, it emphasizes the importance of large datasets and advanced training techniques for achieving high performance in genre classification. The paper "Music Genre Classification Using Convolutional Neural Networks"[8] published in the Journal of Intelligent Systems (2017) explores the use of Convolutional Neural Networks (CNNs) for automatic music genre classification. The authors focus on leveraging CNNs' ability to extract features directly from raw audio data, particularly using spectrograms as input. The study demonstrates that CNNs can effectively capture temporal and spectral patterns in music, leading to improved classification accuracy compared to traditional machine learning methods.

The paper highlights the advantages of CNNs in handling large datasets and complex audio structures, making them a promising approach for genre classification. The paper

"A Feature Selection Method for Music Genre Classification"[9] published in IEEE Transactions on Multimedia (2013) presents a novel feature selection technique to enhance music genre classification accuracy. The authors focus on selecting the most relevant audio features, such as timbre, rhythm, and pitch, using methods like Principal Component Analysis (PCA) to reduce dimensionality and improve model performance. The study demonstrates that effective feature selection significantly enhances the classification accuracy of machine learning models like Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN). The proposed method reduces computational complexity, making music genre classification more efficient and accurate.

The paper "Music Genre Classification Using Spectrogram and Deep Learning"[10] published in the Journal of Computer Science and Technology (2020) investigates the use of spectrograms and deep learning models for music genre classification. The authors utilize spectrograms to visually represent audio signals, allowing Convolutional Neural Networks (CNNs) to capture intricate time-frequency patterns. The study demonstrates that CNNs, when trained on spectrograms, can effectively classify music genres with higher accuracy compared to traditional methods. to This approach leverages the visual structure of audio data improve classification performance, highlighting deep learning's potential in music genre recognition tasks, especially with large datasets.

3 Methodolgy

Dataset The dataset used in this study consists of audio files across multiple genres, each representing a distinct style of music. To classify these songs into their respective genres, we utilized two different approaches to handle the data [4]: Direct Feature Extraction from Audio Files: In this approach, we processed the raw audio files to extract features that characterize the musical attributes of each song [11]. Pre-extracted Features from CSV File: In the second approach, we used a pre-processed CSV file containing features that were already extracted from the audio files. This method allowed us to bypass the computationally expensive process of extracting features during model training and testing [12]. For consistency and efficiency, the final evaluations in this study were conducted using the preextracted features from the CSV file, ensuring uniformity across all models. For the purpose of maintaining consistency, reducing computational overhead, and ensuring fair comparison across different models, the final evaluations and analyses in this study were carried out using the pre-extracted features from the CSV file.

This approach provided a uniform and optimized foundation for implementing and comparing machine learning algorithms, ultimately enhancing the reliability and accuracy of the genre classification results. Feature extraction process 1. Mel-frequency Cepstral Coefficients (MFCCs): Mel-Frequency Cepstral Coefficients (MFCCs) are among the most widely used and effective features for analyzing and classifying audio

signals, especially in tasks involving speech recognition, speaker identification, and music genre classification. They serve as a compact representation of the short-term power spectrum of an audio signal and are designed to mimic the human auditory perception of sound frequencies. Since human hearing is more sensitive to lower frequencies than higher ones, MFCCs transform the frequency scale of an audio signal to the mel scale, which is a perceptual scale that approximates how humans perceive pitch and tone. The computation of MFCCs involves several key signal processing steps. First, the raw audio waveform is divided into small overlapping frames, typically of 20–40 milliseconds in duration, since audio signals are non-stationary and need to be analyzed over short time intervals. Each frame is then passed through a windowing function (such as a Hamming window) to minimize spectral leakage at the boundaries.

After this, the Fast Fourier Transform (FFT) is applied to convert the time-domain signal into its frequency-domain representation, producing a spectrum that reflects the distribution of signal energy across various frequencies. Next, the spectrum is passed through a Melfilter bank, which consists of triangular filters spaced according to the mel scale. This step emphasizes frequencies that are more perceptually relevant to human hearing while reducing the effect of less significant frequencies. The logarithm of the filtered signal is then taken to approximate the human ear's logarithmic response to sound intensity.

Finally, a Discrete Cosine Transform (DCT) is applied to the log-mel spectrum, resulting in the MFCCs. These coefficients represent the overall shape of the spectral envelope, capturing essential timbral characteristics that distinguish one type of sound from another. These coefficients represent the short-term power spectrum of the sound signal and are widely used in speech and music classification tasks. For each audio file, we extracted 40 MFCCs, which were then averaged to obtain a compact representation of the audio signal [13]. 2. 2. Chroma Feature: The Chroma feature, also known as the Chroma vector or Chromagram, is one of the most informative representations of an audio signal when it comes to identifying musical characteristics related to harmony and pitch. It captures the intensity of each of the 12 distinct pitch classes—C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯, and B—of the Western musical scale, regardless of the octave in which they occur. This makes the chroma feature particularly effective for analyzing harmonic and melodic content, as it focuses on the tonal aspects of music rather than its timbral or rhythmic components.

The process of computing the chroma feature begins with transforming the raw audio signal from the time domain to the frequency domain using the Short-Time Fourier Transform (STFT). This transformation allows the analysis of how the signal's frequency content evolves over time. Once the frequency spectrum is obtained, the spectral bins are mapped into 12 categories corresponding to the 12 pitch classes. This mapping is performed in a logarithmic frequency scale, which aligns with how human perception interprets pitch—each octave doubling the frequency of the previous one. Each chroma vector therefore represents the relative intensity of each pitch class present in a given frame of the audio signal. For instance, in a chord containing notes C, E, and G,

the chroma vector will show higher intensities for those three pitch classes, regardless of whether the chord is played in a low or high octave. By aggregating these chroma vectors over time, a chromagram is produced—a twodimensional representation showing how pitch class energy varies throughout the duration of the song. Chroma features represent the distribution of the 12 different pitch classes. Since music is often organized by pitch, this feature captures harmonic content that is important for genre classification [14].

In the first approach, these features were extracted from the raw audio files directly before training the models. However, in the second approach, these features were pre extracted and provided in a CSV file, which allowed for faster training and testing [15]. Data Preprocessing The raw audio data was preprocessed in the following steps: Loading and Resampling: Each audio file was loaded and resampled to a consistent sampling rate (typically 22,050 3. K-Nearest Neighbors (KNN): Hz) using the librosa. load() function. This ensured uniformity across all audio files [12]. Feature Extraction: Using librosa, the features mentioned above were extracted. For MFCCs, a matrix of shape (n_mfcc, T) was generated, where n_mfcc is the number of coefficients and T is the number of time frames. The mean of each coefficient was then calculated to create a fixed-size feature vector [16]. Normalization: All extracted features were standardized to have zero mean and unit variance using StandardScaler from scikit-learn. This step was important for algorithms such as KNN and SVM, which are sensitive to the scale of the input data [17]. Models Overview 1. Random Forest Classifier: Description: Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their predictions. It can handle both categorical and continuous features and is robust against overfitting due to the randomness involved in its construction [16]. Implementation: We used the RandomForestClassifier from the sklearn.ensemble module.

The model was configured with 100 decision trees (n_estimators=100) and was trained using the pre-extracted features [18]. Performance: This model achieved the highest accuracy at 87%, demonstrating strong classification capability across different genres. Its ability to model complex interactions between features likely contributed to this high performance [19]. 2. Support Vector Machine (SVM): Description: SVM is a supervised learning algorithm that finds the optimal hyperplane that maximizes the margin between different classes. SVM is particularly effective for high-dimensional data, such as the feature space generated in this study [16]. Implementation: The SVM model was implemented using the SVC class from sklearn.svm. We used the radial basis function (RBF) kernel to allow the model to handle nonlinear relationships between features. Hyperparameters were tuned using cross-validation [20]. Performance: The SVM model achieved an accuracy of 71%, performing well in genre classification but not as effectively as Random Forest. The relatively lower performance may be attributed to the non-linear nature of the data, which the RBF kernel could only partially capture [21]. 3. K-Nearest Neighbors (KNN): Description: KNN is a non-parametric method that classifies a sample based on the majority vote of its k nearest neighbors in the feature space. While simple to implement, KNN is sensitive to the choice of k and can

struggle with high-dimensional data [16]. Implementation: We used the K Neighbors Classifier from sklearn. neighbors and experimented with different values for k (the number of neighbors) and distance metrics (Euclidean and Manhattan distances). The final model was tuned using grid search cross-validation to find the optimal k [22]. Performance: KNN performed the worst, with an accuracy of 27%. The poor performance is likely due to the high dimensional feature space, where KNN can suffer from the "curse of dimensionality," leading to poor generalization [23]. Evaluation Process

To evaluate the models, we used an 80/20 train-test split, ensuring that 80% of the data was used for training and 20% for testing. The pre-extracted features from the CSV file were used for both training and testing. This approach ensured consistency and fairness in the evaluation of all three models.[24] Training: Each model was trained using the training data, which consisted of the pre-extracted features.[25] Testing: After training, the models were evaluated on the test set, and metrics such as accuracy, precision, recall, and F1-score were calculated.
Accuracy = TP+TN/TP+TN+FP+FN[26], Precision = TP/TP+FP, Recall = TP/TP+FN, and the F-1 score = 2 (Precision * Recall/Precision + Recall). The final comparison of the models was based on accuracy, as shown in the results section.[26]

4 Result

In this study, we have classified genre of songs using audio files. We evaluated the performance of three machine learning models—Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN)—on the task of song genre classification. The evaluation was conducted using features extracted from a pre-processed CSV file, ensuring consistency across all models for comparative analysis. Model Accuracies Random Forest: Random Forest is a powerful and widely used ensemble learning algorithm that combines the predictions of multiple decision trees to improve accuracy, robustness, and generalization. It is based on the principle that a group of weak learners, when combined properly, can form a strong learner. In classification tasks, such as predicting the genre of a song from its audio features, Random Forest helps in achieving high accuracy and stability while minimizing the risks of overfitting that often occur with single decision trees.

At its core, a Random Forest is composed of many independent decision trees, each trained on a random subset of the training data and using a random subset of the input features. This randomness introduces diversity among the trees, ensuring that they do not all make the same errors. When a new data point is introduced, each tree in the forest provides its own prediction, and the final output is determined by a majority vote (for classification) or the average of all tree predictions (for regression). This ensemble approach effectively reduces the variance of the model and improves predictive performance. The training process of a Random Forest involves two main randomization steps: bootstrap sampling and feature selection. Bootstrap sampling means that each

tree is trained on a randomly selected subset of the training data, with replacement, meaning some data points may be repeated while others are omitted. This ensures that each tree sees slightly different data, leading to diversity in the decision boundaries.

The second randomization occurs when splitting nodes in each tree. Instead of considering all features to find the best split, the algorithm selects a random subset of features, which helps to reduce correlation between trees and further enhances the overall model's generalization ability. The Random Forest classifier achieved an accuracy of 87%, making it the most effective model in this study. Random Forest's ensemble learning approach and ability to handle both linear and non-linear relationships between features allowed it to outperform the other classifiers significantly. This model's performance suggests that it effectively captured the complex feature interactions present in the dataset.

Accuracy for CSV-based input: 87.29%
Classification Report:

	precision	recall	f1-score	support
blues	0.87	0.85	0.86	208
classical	0.92	0.98	0.95	203
country	0.74	0.82	0.78	186
disco	0.88	0.84	0.86	199
hiphop	0.91	0.87	0.89	218
jazz	0.84	0.90	0.87	192
metal	0.90	0.95	0.92	204
pop	0.93	0.94	0.93	180
reggae	0.90	0.85	0.87	211
rock	0.85	0.74	0.79	197
accuracy			0.87	1998
macro avg	0.87	0.87	0.87	1998
weighted avg	0.87	0.87	0.87	1998

Fig. 1 Classification report for Random Forest

The image displays a classification report for a machine learning model with an accuracy of 87.29%. It shows precision, recall, F1-score, and support for various music genres such as blues, classical, country, disco, etc. Most genres have high precision and recall, with classical, pop, and metal performing exceptionally well (F1-scores above 0.90). Some genres, like country and rock, have lower F1-scores, indicating more difficulty in predicting these categories accurately. The overall macro and weighted averages of precision, recall, and F1-score are all 0.87, reflecting balanced model

performance across categories. Support Vector Machine (SVM): Support Vector Machine (SVM) is a powerful supervised machine learning algorithm primarily used for classification and regression tasks. In the context of music genre classification, SVM helps distinguish between various musical styles by analyzing and separating patterns within the extracted audio features. It is especially known for its ability to handle complex, high- dimensional datasets effectively and achieve high accuracy even with limited training samples.

The main principle behind SVM is the concept of finding an optimal hyperplane that separates data points of different classes with the maximum possible margin. In a twodimensional space, this hyperplane can be visualized as a line that divides the data into two groups. In higher dimensions, it becomes a hyperplane that separates the feature space based on their respective class labels. The data points that are closest to this hyperplane are known as support vectors, and they play a critical role in defining the position and orientation of the boundary. These points essentially "support" the decision surface, hence the name Support Vector Machine. SVM works by transforming the input data into a higher dimensional space using a technique called the kernel trick. This transformation allows the algorithm to find a linear separation even when the data is not linearly separable in its original space. Some commonly used kernel functions include the linear kernel, polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel. Among these, the RBF kernel is widely used in practical applications as it can effectively handle non-linear relationships by mapping data into an infinite-dimensional space where a separating hyperplane can be easily established. During the training process, SVM attempts to minimize a cost function that balances two objectives: maximizing the margin between classes and minimizing classification errors. This trade-off is controlled by a parameter called C (regularization parameter). A smaller value of C allows a wider margin but tolerates more misclassifications, while a larger C focuses on correctly classifying all training points, potentially leading to overfitting.

Another key parameter, gamma, determines how much influence a single training example has, affecting the curvature of the decision boundary. Fine-tuning these parameters is essential for achieving optimal model performance. One of the key advantages of SVM is its effectiveness in high-dimensional spaces and its robustness to overfitting, especially when the number of features exceeds the number of samples. It performs well on both linear and non-linear classification problems and can be applied to a wide variety of real-world datasets. Additionally, SVM provides a well-defined mathematical framework and often produces highly accurate results even with limited data. However, it also has certain limitations — training can be computationally expensive for very large datasets, and selecting the right kernel function and parameters requires careful tuning and experimentation. The SVM classifier achieved an accuracy of 71%, performing moderately well in classifying the song genres. While it was able to separate genres with reasonable accuracy, SVM's reliance on finding optimal hyperplanes may have limited its performance relative to the Random Forest model, particularly given the non-linear and high- dimensional nature of the audio features.

Accuracy for CSV-based input: 71.34%
Classification Report:

	precision	recall	f1-score	support
blues	0.70	0.68	0.69	208
classical	0.78	0.89	0.83	203
country	0.61	0.65	0.63	186
disco	0.66	0.71	0.68	199
hiphop	0.73	0.70	0.72	218
jazz	0.65	0.71	0.68	192
metal	0.75	0.80	0.78	204
pop	0.76	0.75	0.75	180
reggae	0.70	0.69	0.70	211
rock	0.62	0.58	0.60	197
accuracy			0.71	1998
macro avg	0.70	0.71	0.71	1998
weighted avg	0.71	0.71	0.71	1998

Fig. 2 Classification report for SVM

The image shows a classification report for a model with an accuracy of 71.34%. It includes precision, recall, F1- score, and support for various music genres. The performance is lower across most genres compared to the previous report. Classical and metal perform relatively well (F1-scores above 0.80), while country, rock, and disco show weaker performance (F1-scores below 0.70). The macro and weighted averages for precision, recall, and F1-score are all around 0.71, reflecting lower overall performance and more difficulty in accurately classifying the genres. K-Nearest Neighbors (KNN): The KNN model achieved the lowest accuracy at 27%. This poor performance indicates that KNN was not well-suited for the high-dimensional feature space generated from the audio data. The algorithm's sensitivity to the local distribution of data and its tendency to be affected by noisy features may have contributed to its underperformance.

Accuracy for CSV-based input: 27.13%
Classification Report:

	precision	recall	f1-score	support
blues	0.29	0.22	0.25	208
classical	0.40	0.35	0.37	203
country	0.19	0.25	0.21	186
disco	0.22	0.28	0.24	199
hiphop	0.33	0.29	0.31	218
jazz	0.25	0.18	0.21	192
metal	0.32	0.36	0.34	204
pop	0.24	0.21	0.22	180
reggae	0.27	0.25	0.26	211
rock	0.19	0.14	0.16	197
accuracy			0.27	1998
macro avg	0.27	0.27	0.26	1998
weighted avg	0.28	0.27	0.27	1998

Fig. 3 Classification for KNN

The image shows a classification report for a model with a much lower accuracy of 27.13%. The precision, recall, and F1-scores are significantly low across all music genres, indicating poor model performance. Classical and metal genres still perform relatively better compared to others, but their F1-scores are below 0.40. Genres like country, rock, and jazz have very low F1-scores, around 0.20 or less, suggesting major difficulty in classifying these categories. The macro and weighted averages for precision, recall, and F1-score are all around 0.27, reflecting a general failure in accurate predictions for this dataset.

Comparison and Observation

We initially explored two approaches in this study:

1. Direct Feature Extraction: In this method, features were extracted directly from audio files during model training.
2. Pre-extracted Features: For final testing, we opted to use pre-extracted features from a CSV file to ensure uniformity

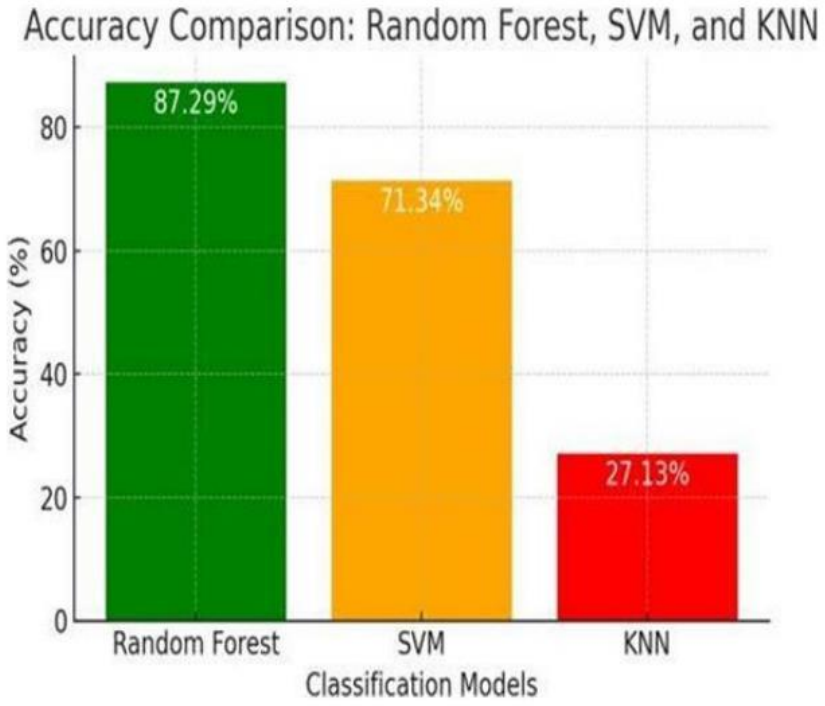

Fig. 4 Accuracy Comparison graph

The results clearly demonstrate that Random Forest outperformed both SVM and KNN, likely due to its robust ensemble learning mechanism. On the other hand, KNN struggled significantly, likely due to the complexity of the feature set and the high-dimensional space it operates in. SVM performed well but was not able to match the versatility of Random Forest in handling the intricate relationships between feature.

5 Conclusion

Music genre classification has witnessed remarkable progress over the years, largely driven by advancements in machine learning, deep learning, and digital signal processing. What began as a simple task of grouping songs based on manually selected features has now evolved into a complex and highly efficient process powered by intelligent algorithms and largescale data analytics. The core goal of genre classification is to automatically categorize audio signals into predefined musical genres such as rock, pop, jazz, classical, or hip-hop, based on their inherent characteristics like rhythm, timbre, pitch, and spectral content. This area of research plays a significant role in the modern music industry, particularly in applications like automated playlist generation, music recommendation systems, and organization of large digital music libraries. In this study, machine learning algorithms such as Support Vector Machine (SVM),

Random Forest, and K-Nearest Neighbors (KNN) were employed to perform genre classification based on pre-extracted audio features. Each of these models brings unique strengths and methodologies to the problem, offering different perspectives on how music data can be analyzed and interpreted. Through these approaches, the study explored not only the accuracy of genre prediction but also the impact of feature representation, dataset quality, and computational efficiency on model performance. The first step in any audio classification task is effective feature extraction. Raw audio signals are complex and continuous, making them difficult to process directly using machine learning algorithms. Therefore, meaningful features that capture the essential aspects of sound are extracted. Commonly used features include Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, spectral contrast, zero-crossing rate, tempo, and spectral centroid. These features collectively describe the timbral, harmonic, and rhythmic properties of the music, which are crucial for distinguishing between genres. In the present study, both direct feature extraction from audio files and the use of preextracted features from a CSV file were considered. Using the pre-extracted dataset ensured consistency, reduced computation time, and provided a uniform foundation for training and testing the machine learning models. The Support Vector Machine (SVM) algorithm played an important role due to its strong theoretical foundation and proven performance in classification tasks. SVM's ability to find an optimal hyperplane that maximizes the margin between different classes made it highly effective in separating overlapping musical patterns. The kernel trick further enhanced its flexibility, enabling it to handle nonlinear relationships present in the audio data. By mapping features into higher-dimensional spaces, SVM could identify subtle boundaries between genres such as pop and electronic or jazz and blues. Moreover, its resistance to overfitting and robustness with high-dimensional data made it an excellent baseline model for genre classification. The Random Forest algorithm offered a different yet complementary perspective. As an ensemble learning technique, it combined the outputs of multiple decision trees to improve predictive accuracy and stability. Each tree in the forest was trained on a random subset of the dataset, ensuring diversity and minimizing bias. Random Forest's ability to handle both categorical and continuous data, along with its feature importance analysis, made it a valuable tool for understanding which audio characteristics most strongly influenced genre prediction. For example, features like MFCCs and spectral roll-off were often found to play a significant role in distinguishing between rhythm-intensive genres like rock and tempo-based genres like classical or jazz. Although computationally more demanding, Random Forest demonstrated strong generalization performance and reliable accuracy, making it a suitable choice for this application. Another algorithm utilized was K-Nearest Neighbors (KNN), a simple yet powerful non-parametric method based on the concept of similarity. KNN classifies a data point by analyzing the majority class among its nearest neighbors in the feature space. It does not involve any training phase, which makes it computationally lightweight in model creation, though prediction can be slower for large datasets. Despite its simplicity, KNN performed remarkably well for music genre classification, especially when properly tuned with an optimal number of neighbors and appropriate distance metrics such as Euclidean or Manhattan distance. However, KNN's sensitivity to feature scaling and dataset imbalance was noted as a limitation, highlighting the need

for normalization and data preprocessing before applying the algorithm. A key factor in improving the performance of all these models lies in the representation of musical data. Traditional feature extraction methods have now been supplemented by more advanced techniques, including timefrequency representations like spectrograms, mel-spectrograms, and chromagrams. These visual representations of audio signals allow models to capture temporal and spectral patterns more effectively, mirroring how humans perceive sound. Furthermore, the integration of deep learning architectures, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), has revolutionized this domain by automating the feature extraction process. CNNs, for instance, can directly learn from spectrograms and identify patterns in frequency and time domains, providing superior accuracy compared to traditional handcrafted features

References

1. Xu, C., Maddage, N. C., Shao, X., Cao, F., & Tian, Q.: Musical genre classification using support vector machines. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03). 5, V-429) (2003)
2. Choi, Keunwoo, György Fazekas, Mark Sandler, and Kyunghyun Cho: Convolutional recurrent neural networks for music classification. In: IEEE International conference on acoustics, speech and signal processing (ICASSP), 2392-2396. (2017)
3. Long, Mingtao, Luyao Hu, and Fubao Jin: Analysis of main characteristics of music genre based on PCA algorithm. In: 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 101-105. IEEE, (2021).
4. Tzanetakis, George, and Perry Cook: Musical genre classification of audio signals. IEEE Transactions on speech and audio processing **10**(5), 293-302, (2002)
5. Hu, Xiao, and J. Stephen Downie: Improving mood classification in music digital libraries by combining lyrics and audio. In: Proceedings of the 10th annual joint conference on Digital libraries, 159-168. (2010)
6. Costa, Y. M. G., Oliveira, L. S., Silla Jr, C. N., & Koerich, A. L.: Music genre classification using LBP textural features. Signal Processing, **92**(11), 2723–2737 (2012)
7. Elbir, Ahmet, and Nizamettin Aydin: Music genre classification and music recommendation by using deep learning. Electronics Letters **56**(12) 627-629. (2020)
8. Dieleman, Sander, and Benjamin Schrauwen. End-to-end learning for music audio. IEEE international conference on acoustics, speech and signal processing (ICASSP) (2014)
9. Li, Tao, Mitsunori Ogihara, and Qi Li: A comparative study on content-based music genre classification. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, 282-289. (2003)
10. Oramas, S., Nieto, O., Sordo, M., & Serra, X.: A deep multimodal approach for cold-start music recommendation. In: Proceedings of the 2nd workshop on deep learning for recommender systems, 32-37 (2017).
11. Ellis, D. P. (2007). Classifying music audio with timbral and chroma features. (2007)
12. McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto: librosa: Audio and music signal analysis in python." SciPy **2015**(7), 18-24 (2015)

13. Logan, Beth: Mel frequency cepstral coefficients for music modeling. In: Ismir, vol. **270**(1), 11. (2000)
14. FUJISHIMA, T.: Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music. CCRMA. (1999).
15. Han, Yoonchang, Jaehun Kim, and Kyogu Lee: Deep convolutional neural networks for predominant instrument recognition in polyphonic music. IEEE/ACM Transactions on Audio, Speech, and Language Processing **25**(1), 208-221 (2016).
16. Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al.: Scikit-learn: Machine learning in Python. The Journal of machine Learning research **12**, 2825-2830 (2011)
17. Breiman, Leo: Random forests. Machine learning **45**(1), 5-32 (2001)
18. Liaw, Andy, and Matthew Wiener: Classification and regression by randomForest. R news **2**(3), 18-22 (2002)
19. Biau, Gérard: Analysis of a random forests model. The Journal of Machine Learning Research **13**, 1063-1095. (2012).
20. Cortes, Corinna, and Vladimir Vapnik: Support-vector networks. Machine learning **20(**3) 273-297 (1995).
21. Schölkopf, Bernhard, and Alexander J. Smola: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, (2002)
22. Cover, Thomas, and Peter Hart: Nearest neighbor pattern classification. IEEE transactions on information theory **13**(1), 21-27 (1967)
23. Beyer, Kevin, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft: When is "nearest neighbor" meaningful?. In: International conference on database theory, Berlin, Heidelberg: Springer Berlin Heidelberg, 217-235 (1999)
24. Kohavi, Ron.: A study of cross-validation and bootstrap for accuracy estimation and model selection. Ijcai, vol. **14(**2), 1137-1145 (1995)
25. Caruana, Rich, and Alexandru Niculescu-Mizil: "An empirical comparison of supervised learning algorithms, in: Proceedings of the 23rd international conference on Machine learning. (2006)
26. Sokolova, Marina, and Guy Lapalme: "A systematic analysis of performance measures for classification tasks." Information processing & management **45**(4), 427-437. (2009)
27. Goodfellow, I., Bengio, Y., & Courville, A.: Deep Learning. MIT Press. (2016)