

## Article

# Large-Scale Music Genre Analysis and Classification Using Machine Learning with Apache Spark

Mousumi Chaudhury , Amin Karami  and Mustansar Ali Ghazanfar

Computer Science and Digital Technologies, University of East London, London E16 2RD, UK

\* Correspondence: [chaudhurmousumi@gmail.com](mailto:chaudhurmousumi@gmail.com)

**Abstract:** The trend for listening to music online has greatly increased over the past decade due to the number of online musical tracks. The large music databases of music libraries that are provided by online music content distribution vendors make music streaming and downloading services more accessible to the end-user. It is essential to classify similar types of songs with an appropriate tag or index (genre) to present similar songs in a convenient way to the end-user. As the trend of online music listening continues to increase, developing multiple machine learning models to classify music genres has become a main area of research. In this research paper, a popular music dataset GTZAN which contains ten music genres is analysed to study various types of music features and audio signals. Multiple scalable machine learning algorithms supported by Apache Spark, including naïve Bayes, decision tree, logistic regression, and random forest, are investigated for the classification of music genres. The performance of these classifiers is compared, and the random forest performs as the best classifier for the classification of music genres. Apache Spark is used in this paper to reduce the computation time for machine learning predictions with no computational cost, as it focuses on parallel computation. The present work also demonstrates that the perfect combination of Apache Spark and machine learning algorithms reduces the scalability problem of the computation of machine learning predictions. Moreover, different hyperparameters of the random forest classifier are optimized to increase the performance efficiency of the classifier in the domain of music genre classification. The experimental outcome shows that the developed random forest classifier can establish a high level of performance accuracy, especially for the mislabelled, distorted GTZAN dataset. This classifier has outperformed other machine learning classifiers supported by Apache Spark in the present work. The random forest classifier manages to achieve 90% accuracy for music genre classification compared to other work in the same domain.



**Citation:** Chaudhury, M.; Karami, A.; Ghazanfar, M.A. Large-Scale Music Genre Analysis and Classification Using Machine Learning with Apache Spark. *Electronics* **2022**, *11*, 2567. <https://doi.org/10.3390/electronics11162567>

Academic Editors: Gemma Piella and Rui Pedro Lopes

Received: 22 June 2022

Accepted: 14 August 2022

Published: 17 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** music genre; Apache Spark; PySpark; machine learning; exploratory data analysis

## 1. Introduction

The digitalization of music has been widely embraced by a plethora of online music organizations in recent years. These organizations run different online [1,2] music channels and distribute entertainment services to their clients [3]. In addition, they group similar musical tracks, assign a tag, and deliver it to their clients. This grouping of similar musical tracks enhances clients' understanding and interest in musical libraries. Clients can benefit from the analysis and categorization of musical tracks. As a result, the business demand and profitability margin can be achieved by many music organizations. Analysing a large volume of musical datasets for retrieving music information has become an emerging field of research areas recently [4]. A categorical label (genre) is assigned to each music piece to identify the kind of music. The user is responsible for assigning the genre tag to a particular song according to their judgement of music by acknowledging a set of music features. This acknowledgement from users comes in a form of a huge music database for music organizations. Due to a large number of online music collections, the categorization of music genres is important for music organizations to search, retrieve, recognize, and

recommend excerpts of music to their clients [5]. Music genre recognition (MGR) was first investigated by Cook and Tzanetakis (2002) [4]. The authors focused on the audio (music) pattern recognition job in the domain of music information retrieval (MIR). This is considered the first research on maintaining a large-size music dataset.

Music genre recognition (or classification) contains several phases. The initial phase is to extract a set of important features from raw music signals and implement feature selection methods on these raw audio signals. In addition, the analysis of several characteristics of the waveforms of music signals is an essential phase to understand audio signals [6,7]. Much research implements the extraction of segment-level, frame-level, and song-level features of audio signals. The spectral characteristics of any audio signal are defined by frame-level features such as spectral roll-off, spectral centroid, and Mel frequency cepstral coefficients and are computed from short time frames. The statistical measures of an audio segment that is composed of several frames calculate segment-level features. Song-level features, such as rhythmic information, tempo, and pitches, define music tracks in user-understandable formats. Various types of music (or audio) feature extraction procedures are adopted by researchers [6,7]. Some researchers use Mel frequency cepstral coefficients as the classification criteria, while other researchers prefer to use tempo and pitch features. Besides that, spectrograms computed from the audio signal are extracted in research for the classification of music genres [8,9].

Moreover, classifying the specific genre of music is the initial phase in recommendation and many music-based applications. In the recent years, machine learning models have been used to classify the kind of music for improving and recommending the music listening experience of the user [10,11]. Data science helps to define the steps to prepare the data before using it to train a machine learning classifier [12]. The steps for preparing data include cleaning and aggregating the raw data. Machine learning is a subdivision of artificial intelligence that can learn any specific domain features from input data and can solve a problem related to the trained domain. It uses science such as maths and statistics to learn the pattern of features from the input data themselves. When implemented with a music dataset, a machine learning model can learn several features of music and identify them into groups of similar music. This enables a user to search for a similar song according to their preference. Thus, online music organizations can grow their business by satisfying their clients. Recent research shows that machine learning classifiers, such as artificial neural network, convolutional neural network, decision tree, logistic regression, random forest, support vector machine, and naïve Bayes, can perform better in the domain of music genre classification following the supervised learning method [13]. However, it is difficult to compute machine learning predictions with a large-scale music dataset, as the model training duration and computational cost increase extensively at a higher rate [14]. This has been a major drawback while developing an application with a massive amount of music datasets in recent years. It is a challenge of scalability for machine learning algorithms with large-scale datasets [15,16]. A distributed computing framework called Apache Spark was introduced to overcome this scalability problem. Apache Spark processes data in parallel using a directed acyclic graph (DAG) engine supporting cyclic data flow across multiple nodes, which makes machine learning computations faster for massive datasets [17].

Many researchers have contributed several feature extraction methods and machine learning classifiers to achieve numerous results in music information retrieval research (MIR) and music genre classification with massive datasets in recent years. However, one major gap in those contributions is the lack of technology to reduce the training duration and cost of computing machine learning predictions. In the real world, massive amounts of music collections are analysed and classified into appropriate music genres to develop applications such as music classification, speech recognition, music information extraction systems, automatic music tagging, and many more. Therefore, it is essential to improve the speed of data processing to run and update any application. Moreover, a comparison between an ensemble learning classifier such as random forest supported by Apache Spark and other machine learning classifiers in the domain of music genre classification can

be considered as an additional gap with earlier research. The random forest which was developed by Breiman (2001) [18] is considered the best classifier following the supervised learning method. This research paper aims to analyse the statistical features of a large number of music datasets and classify music into groups of similar music (music genres) using the ensemble learning classifier random forest supported by Apache Spark. In addition, the contributions of this paper are enumerated below:

1. The statistical features of music from a big-size single dataset GTZAN are analysed and visualised.
2. An ensemble learning classifier random forest is developed and implemented to classify the types of music (music genres).
3. An in-memory distributed computing framework Apache Spark is used to process the data in parallel to reduce the duration of machine learning predictions without computational cost.
4. Multiple machine learning classifiers supported by Apache Spark, such as naïve Bayes, decision tree, and logistic regression, are also implemented to compare the performance efficiency with random forest for the classification of music genres.

The remaining part of this paper is categorized into six sections. In Section 2, related pieces from the literature are reviewed to acknowledge the methodologies followed by researchers in recent years. In Section 3, a description of the dataset and implemented research methodologies are discussed. Section 4 demonstrates the experimental setup, the outcome of the research, and discussions on the outperformance of the developed classifier. Section 5 highlights the further analysis and discussions on key findings with pieces from the literature. Finally, the conclusion of this research paper is drawn in Section 6, and future insights are highlighted in Section 7.

## 2. Literature Review

Music genre classification and music information retrieval have been actively investigated over the last decade. Different modern machine learning technologies have been adopted in this research field. Various reviews have been presented to analyse appropriate music features and classification algorithms that are explored in the domain of music genre recognition. Wibowo and Wihayati (2022) [19] proposed a deep learning approach for the classification of music genres. The authors explored the GTZAN dataset with ten foreign music genres. The deep learning model obtained a classification accuracy above 90%. In the next stage, an additional dataset as popular as the dangdut music genre was added for the identification of the dangdut music genre among Western music genres. It was observed that the performance of the deep learning model with the dangdut music genre decreased to around 76%. The result highlighted that dangdut music is different from other foreign music genres, but few music genres such as jazz and pop were identified. Puppla and Muvva (2021) [20] designed a convolutional neural network using a deep learning approach for the training and classification of music genres from the GTZAN dataset. The Mel frequency cepstral constant (MFCC) feature vector was extracted and utilized for the classification process. The dataset was divided into 60% for training and 40% for testing purposes. The training and testing accuracies were 97% and 74%, respectively, with this approach. In the same year, a case study of a parallel deep neural network in the domain of music genre recognition was presented by Yuan and Zheng (2021) [21]. The authors proposed a deep learning model based on a hybrid framework in which the convolutional neural network and the recurrent neural network were positioned parallelly. The model was trained and evaluated using a popular dataset known as the Free Music Archive (FMA). The performance accuracy of the model was 88% on the FMA dataset. Furthermore, the model was tested on a curated dataset of fifteen songs and managed to classify 11 songs out of 15 songs. Another hybrid architecture consisting of the parallel convolutional neural network (CNN) and bidirectional recurrent neural network (Bi-RNN) blocks was implemented by Feng and Liu (2017) [22]. This architecture was proposed to evaluate the robustness of the extracted features and to improve the performance of

the deep learning model. The authors focused on extracting the spatial features of music through CNN. The experimental result showed that the designed CNN with Bi-RNN had 92% accuracy for music genre classification.

Besides a convolutional neural network using a deep learning approach and a recurrent neural network, researchers also investigated the applications of different machine learning algorithms, such as K-nearest neighbor (KNN), support vector machine (SVM), and artificial neural network (ANN), in the domain of music genre classification. Kumar and Chaturvedi (2020) [23] elaborated an audio classification approach using an artificial neural network. The authors emphasized more audio feature extraction approaches, such as chroma- or centroid-based features, Mel frequency cepstral coefficients (MFCCs), and linear predictive coding coefficients (LPCCs), to understand the behaviour of the audio signal. An efficient neural network classifier was used for the classification of audio signals with a high accuracy rate. Alternatively, Kobayashi and Kubota (2018) [24] proposed a different approach to audio feature extraction and classification. The authors introduced unique data preprocessing steps to extract musical feature vectors. At the initial stage, the normalization method was applied to input audio signals to decompose signals into many signals with different resolutions using the undecimated wavelet transform (UWT). Each signal was divided into multiple local frames. Some basic statistics such as correlations were calculated for all the signals that were known as sub-band signals and were integrated into a single feature vector. The GTZAN dataset was used for the investigation of audio signals. In the next stage, the authors used the support vector machine classifier for the classification procedure with the best accuracy of 81.5%. Moreover, the same classifier was applied for the classification of music genres by Chaudary and Aziz (2021) [25]. In the data preprocessing stage, the empirical mode decomposition (EMD) process was used to preprocess the audio signals into several intrinsic mode functions (IMFs). The empirical mode decomposition (EMD) is an analysis method to decompose a signal in the time domain. Therefore, the appropriate region of the audio dataset was extracted by using the empirical mode decomposition (EMD) method, and the time and frequency domain features were selected for the linear support vector machine classifier. The suitable feature extraction with several methods was highlighted by the author in this article to decrease the cost of computation. After the data preprocessing stage, the machine learning algorithm support vector machine was used for its excellent performance in the classification of music genres. The blues, classical, metal, hip-hop, and pop genres were selected for the classifier out of ten classes to achieve the best accuracy. Pelchat and Gelowitz (2020) [26] implemented a neural network for the music genre classification. The dataset was divided into 70% training subset, 20% validation subset, and 10% testing subset. The performance accuracy of the neural network was 85%. Rong (2016) [27] proposed a different approach to the audio classification method using a machine learning process. In the first step, the author demonstrated the four layers of audio data: audio shot, audio frame, audio clip, and audio high-level semantic unit. In the second step, the zero-crossing rate, short-time energy, and Mel frequency cepstral coefficient features of the audio dataset were extracted and converted to the equivalent feature vector. In the final step, the support vector machine classifier with a Gaussian kernel was implemented for the music genre's classification. The investigation showed that the proposed method managed to achieve a higher classification rate.

A spectacular approach was adopted by Xavier and Thirunavukarasu (2017) [28]. The authors implemented an ensemble learning distributed approach to recognize protein secondary structures. The investigation revealed that the efficiency of the ensemble approach for the classification of protein secondary structures was better with a distributed environment such as Apache Spark. A song recommender system based on a distributed scalable big data framework was proposed by Kose and Eken (2016) [29]. The authors applied the Word2vec algorithm in Apache Spark to produce a playlist reflecting an end-user's preferences. In addition, an exploratory teaching program by Eken [30] highlighted that the analysis of massive datasets was very useful in improving any company's analysing

process. Zeng and Tan (2021) [31] developed a large-scale pretrained model MusicBERT for four music understanding tasks, including melody completion, accompaniment suggestion, genre classification, and style classification. Mehta and Gandhi (2021) [32] compared four transfer learning architectures, Resnet34, Resnet50, VGG16, and AlexNet, for music genre classification.

Therefore, it was highlighted in recent pieces of literature that different machine learning models, such as CNN, ANN, KNN, and SVM, with suitable combinations of extracted features in the domain of music genre classification have achieved the best performance accuracy. A few pieces from the literature emphasize the comparative analysis of multiple machine learning algorithms on different audio datasets in the domain of music genre classification. Ignatius Moses Setiadi et al. (2020) [33] implemented a support vector machine with a radial kernel base function (RBF), naïve Bayes, and K-nearest neighbor on the Spotify music dataset for the classification of music genres. The author applied the chi-square method to filter important features, as the dataset had 26 genres with 18 features each. The training process was increased after the feature filtering process. After several investigations, the support vector machine classifier with RBF had achieved the best classification accuracy of 80%. Kumar and Sowmya (2016) [34] proposed a detailed comparative study to classify music genres by using different classification algorithms. The author implemented a series of machine learning algorithms such as K-nearest neighbor, logistic regression, support vector machine, recurrent neural network, and decision tree using the GTZAN dataset. The Mel frequency cepstral coefficients (MFCCs) feature and fast Fourier transform (FFT) method were normalized for the investigation procedure. The result highlighted that the support vector machine and the logistic regression had the highest classification accuracy compared to other classifiers utilized in the study.

Furthermore, some works in the literature extensively discuss the utilization of multiple music datasets for the classification of music genres. The GTZAN, Free Music Archive (FMA), Million Songs Dataset, and Spotify's huge database were thoroughly analysed and applied in the domain of music genre classification. Khasgiwala and Tailor (2021) [35] used the Free Music Archive (FMA) dataset for the classification of music genres. The authors implemented the recurrent neural network and the convolutional neural network. However, a model with a different approach had been designed to improve the performance of the convolutional neural network. The author investigated Mel-frequency cepstral coefficients (MFCC) features of songs on these models.

In the current year (2022), researchers are still analysing several features of music datasets and applying preprocessing techniques to extract features from actual datasets. The convolutional neural network using a deep learning approach and traditional neural network are used for the music genre recognition process. In addition, Singh and Biswas (2022) [36] explained a contradictory approach to the robustness of several music features on different deep learning models for music genre classification. The authors stated that a machine learning model needs large-scale training data to generalize better results with testing data. The widely used musical and non-musical features were assessed for robustness in the article. Selected features were extracted, and the performances of multiple deep learning models were evaluated for the music genre recognition task. The GTZAN, Hindustani, Carnatic, and Homburg music datasets were used to train and validate the deep learning models. The investigation of various music features revealed the robustness of these datasets. In addition, the previous year, Folorunso and Afolabi (2021) [37] implemented multiple machine learning models on the ORIN dataset to classify five music genres. The ORIN dataset consists of 478 Nigerian songs from five genres: fuji, highlife, juju, waka, and apala. The extreme gradient boosting (XGBoost) classifier was the best classifier among other classifiers that were implemented on the ORIN dataset.



The review of the different kinds of literature in Table 1 shows that modern machine learning methods, such as the convolutional neural network, artificial neural network, and recurrent neural network, are widely used in the domain of music genre classification for multiple large-scale datasets. However, it is necessary to preprocess or reduce important image features to proceed with these classifiers to reduce the duration of model training and computational cost. Researchers are still implementing various methods of data preprocessing to reduce the number of training features and to increase the processing time with the reduced features. Kang and Gang (2021) [38] implemented a few feature preprocessing steps, such as computing short-term Fourier transformation, converting images from wave signal to digital signal, and reducing the number of training samples, to increase the duration of training time and improve the performance accuracy of the machine learning models. The usage of modern machine learning classifiers and image processing methods increases the computational cost. Apache Spark also supports a scalable, platform-independent MLlib library which has common algorithms such as classification, regression, clustering, and collaborative filtering [39]. Training a machine learning model using a convolutional neural network using a deep learning approach and a traditional neural network not only increases training time but also increases computational cost. The combination of Apache Spark and machine learning algorithms is used to experiment several real-life problems such as banking data analysis and performance prediction.

**Table 1.** A summary of literature review on the datasets and machine learning classifiers.

Reference	Years	Datasets	Machine Learning Classifiers
[19,36]	2022	GTZAN, Carnatic, Hindustani, and Homburg	Convolutional neural network with deep learning and artificial neural network
[20,21,25]	2021	GTZAN, FMA, and ORIN	Convolutional neural network, recurrent neural network, support vector machine, and extreme gradient boosting
[23,26]	2020,2019	GTZAN and Spotify	Artificial neural network, support vector machine, K-nearest neighbor, and naïve Bayes
[24]	2018	GTZAN	Support vector machine
[22]	2017	GTZAN	Convolutional neural network and bidirectional recurrent neural network
[27]	2016	GTZAN	Support vector machine classifier with a Gaussian kernel, logistic regression, K-nearest neighbor, decision tree, and recurrent neural network

In contrast to earlier outcomes in the domain of music genre classification, however, no evidence of computing statistical features for a complete big-size dataset and calculating machine learning predictions using the ensemble learning algorithm from the machine learning library of Apache Spark has been introduced. In the era of big data, the lack of appropriate strategies to process large-scale datasets when calculating the predictions of machine learning and computational cost makes for a challenging job. In this research paper, the solution to this challenging task is highlighted to complete the interval between earlier contributions. Therefore, an open-source, in-memory, distributed parallel computing framework for Apache Spark is introduced in this paper to overcome this scalability problem of computing large-scale datasets. The implementation of the ensemble learning algorithm for the prediction of music genres is another contribution of this paper. The combined areas of big data and machine learning on scalability are highlighted in the present work. In addition, a comparative analysis of multiple machine learning classifiers in the domain of the classification of music genres is provided.

### 3. Methodology

The methodology section of this paper is categorized into seven steps. The first step, Section 3.1, is to choose an appropriate music dataset for the investigation of recognizing music genres. Any public dataset in the domain of music genre classification can be chosen for this investigation. The second step, Section 3.2, is to analyse the dataset before processing any further steps with it. The third step, Section 3.2.2, is to perform dataset preprocessing, where the dataset is cleaned. The fourth step, Section 3.3, is feature selection, where necessary statistical features are investigated and selected to improve the accuracy of the chosen machine learning model. The fifth step, Section 3.4, is to build and train the random forest classifier for the classification of music genres into ten different classes. The sixth step, Section 3.4, is for applying hyperparameter optimization techniques to improve the performance efficiency of the machine learning classifier. Finally, the verification and evaluation of the machine learning classifier, Sections 3.5 and 3.6, are conducted on the testing dataset using metrics. A benchmark dataset, which is very popular in the domain of music genre classification, is investigated in this paper. Multiple machine learning classifiers from Apache Spark's machine learning library are experimented on the dataset for music genre classification. The best performing classifier is trained to enhance the performance of recognizing music genres from a complete dataset. Different hyperparameters, which are implemented in this paper, are also discussed. The complete investigation procedure of this paper is highlighted in a flow-diagram in Figure 1.

#### 3.1. Description of the Dataset

The primary step of the investigation process is to collect an appropriate dataset for music genre classification. Many researchers have used multiple music datasets for music information extraction and music genre classification in recent years. According to them, the well-known domain-specific GTZAN dataset is the most effective dataset to explore for the analysis and classification of music genres. The dataset is publicly accessible through the Kaggle. It is widely used in research for the prediction of complex genres of music. The GTZAN dataset was first proposed by Cook and Tzanetakis (2002) [4] for music signal processing as stated by Elbir and Bilal Çam (2018) [40]. The dataset contains many features. Each feature contains mean and variance across playtime. It also contains a total of 10,990 music excerpts scattered across ten classes or genres. The classes are blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock; see Figure 2 (GTZAN Dataset—Music Genre Classification, available online: <https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification> (accessed on 10 February 2022)). Each class contains 30 s long music excerpts in .wav format with a sample rate of 22,050 Hz. The music excerpts are split into 3 s audio files which are stored in another CSV file. The dataset also contains a visual representation of each audio file. It is an example of a structured categorical dataset because all numerical features are grouped into a label or category. The dataset is widely utilized in the research in the field of music information retrieval (MIR). Table 2 demonstrates the various music features and representations.

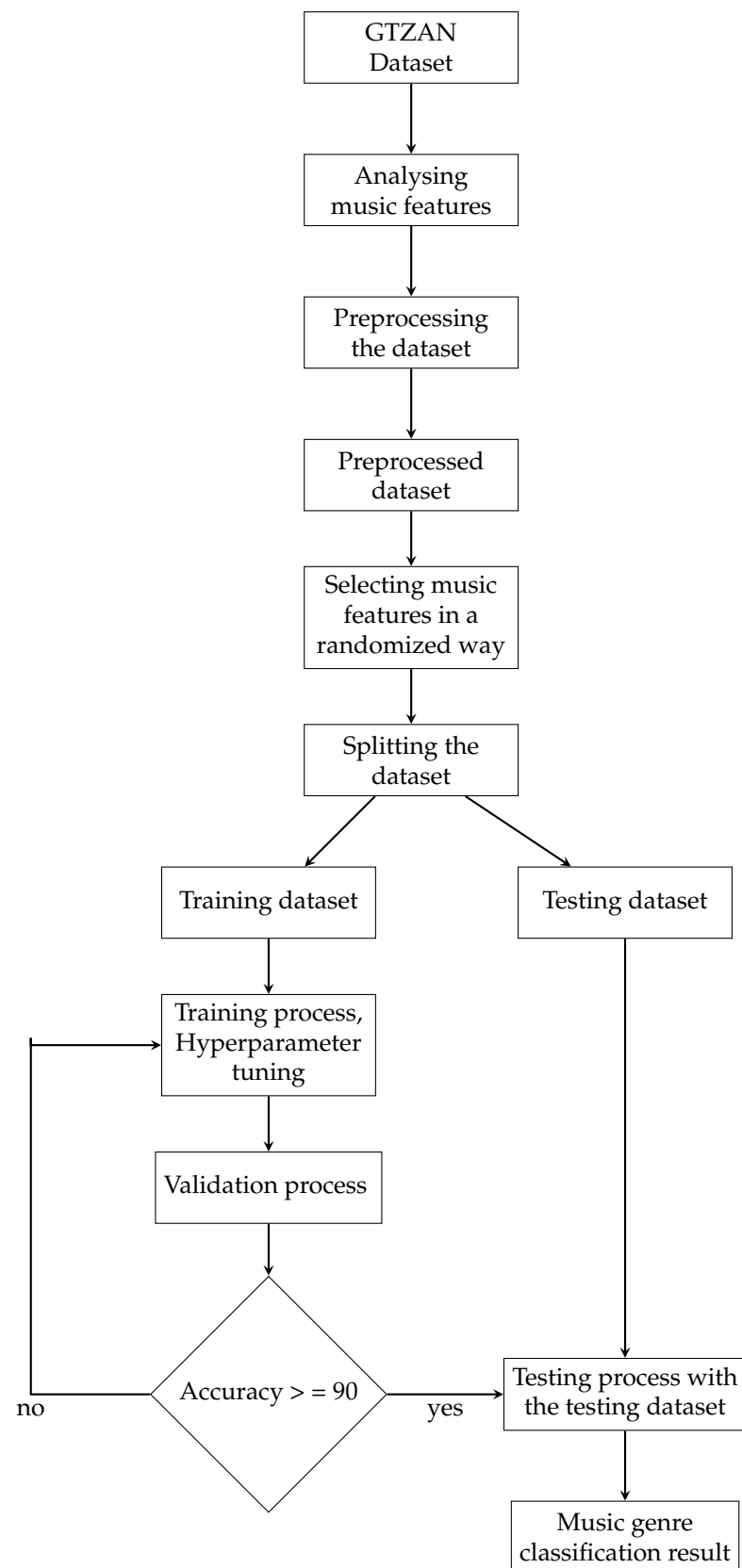


Figure 1. Flow of the investigation process.



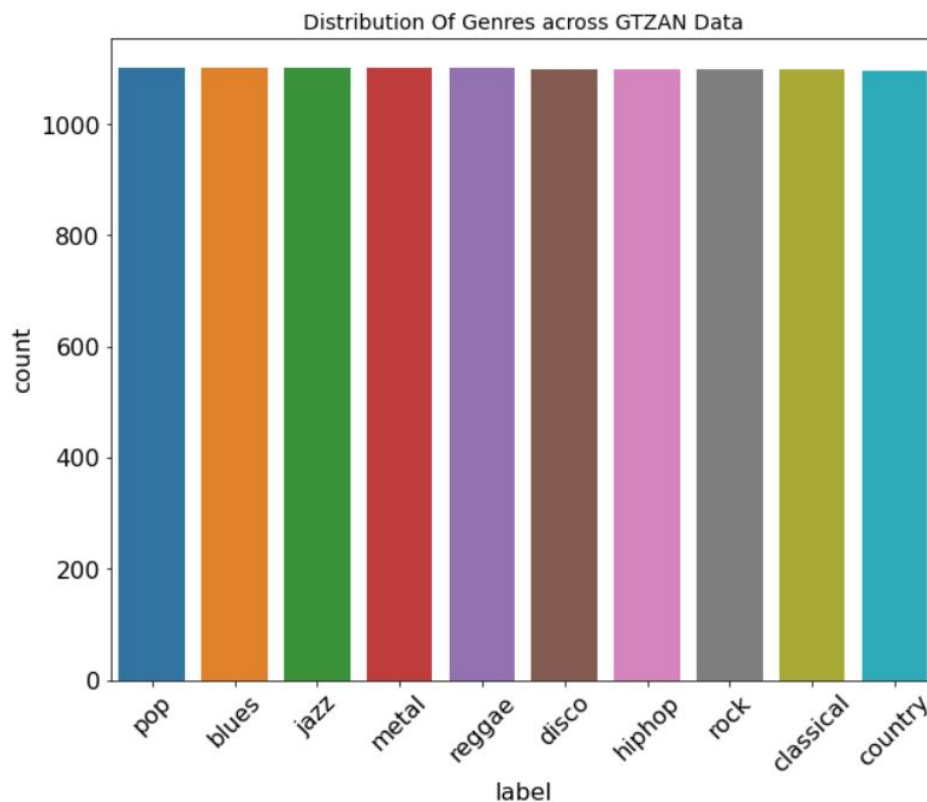


Figure 2. Data distribution of the GTZAN dataset.

Table 2. Music features in the GTZAN dataset.

Music Feature Category	Music Features	Representation
Magnitude-based	Spectral bandwidth Spectral centroid Spectral roll-off Mel frequency cepstral coefficients (MFCCs) 1 to 20	Mean and variance
Tempo-based	Tempo Root mean square energy	Single value Mean and variance
Pitch-based	Zero-crossing rate	Mean and variance
Chordal progression	Harmony	Mean and variance
Chroma-based	Chroma STFT	Mean and variance
Human-perception-based	Perceptual	Mean and variance
-	Length	Single value

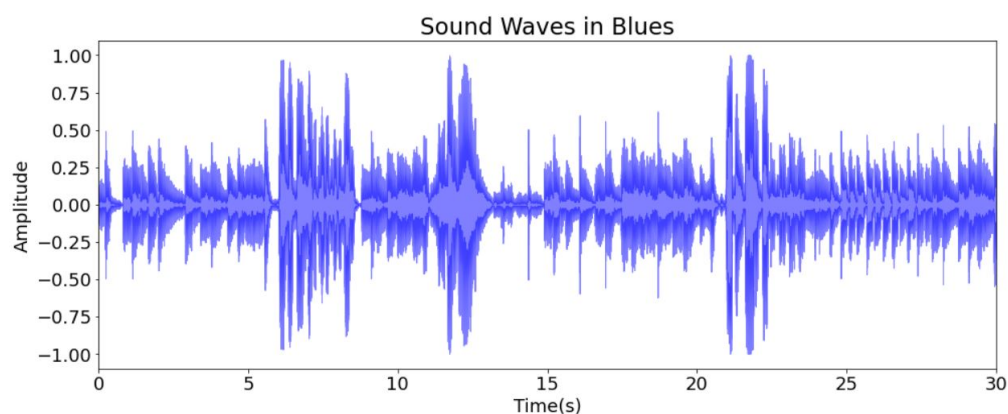
### 3.2. Music Feature Analysis

Studying the GTZAN dataset and acknowledging categories, the statistical measurements of musical features, and time and frequency domain features is essential in order to solve the music genre classification problem. Therefore, it is very important to understand the relationship among the features of the music dataset to perform any processing method on it. The GTZAN dataset contains 100 music excerpts for each class, which is not sufficient to achieve a good classification accuracy. Therefore, a complete dataset with more audio excerpts is analysed in this paper to increase the number of training and testing samples and have the classifier achieve a good classification accuracy. The main aim of this paper is to observe all music features of the GTZAN dataset and to select all statistical features to

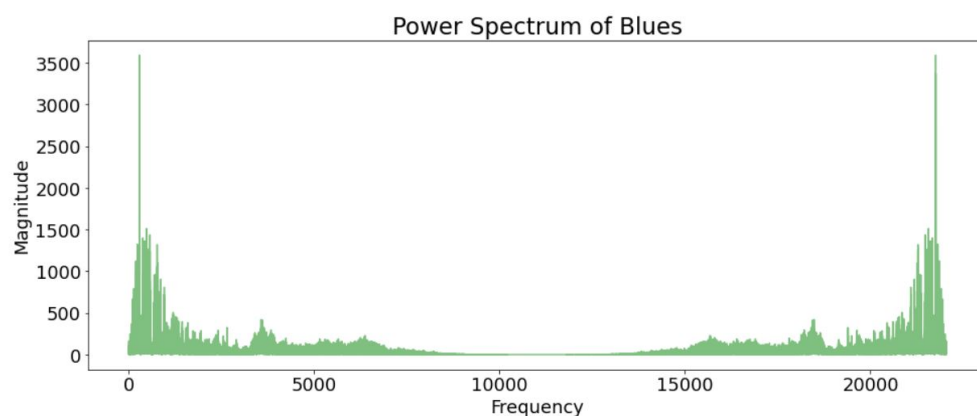
build a machine learning model to classify music genres. Therefore, music feature analysis is conducted in two ways.

### 3.2.1. Analysis of Time and Frequency Domain Features

Sound is expressed in the form of audio signals, having properties such as bandwidth, decibel, and frequency. These audio signals can be digital signals or analog signals. Analog signals have continuous values for time and amplitude. On the other hand, digital signals are sequences of discrete values where data points can only take a finite number of values. Analog signals can be converted to digital signals by the quantization and sampling method. Sampling is a process of sampling data points across a sound wave at a specific point in time. Alternatively, an audio signal is converted from a time domain, Figure 3, to a frequency domain, Figure 4, by using the Fourier transform. A visual demonstration of the spectrum of the frequencies of sound with time is known as a spectrogram. The spectrogram is used to visualise the input music signal from the time domain to the frequency domain.



**Figure 3.** Blues music signal in the time domain.

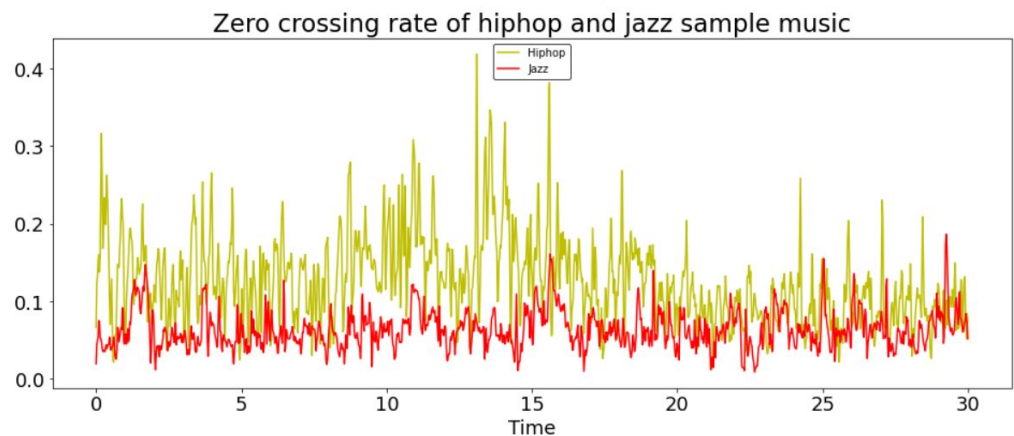


**Figure 4.** Blues music signal in the frequency domain.

Moreover, the numerous features of music signals are investigated in this paper to understand the nature of an audio signal. These features are divided into two categories.

Time domain features:

(a) Zero-crossing rate—The zero-crossing rate is the rate at which a signal changes from a positive value to zero to a negative value or from negative value to zero to a positive value; see Figure 5.



**Figure 5.** Zero-crossing rate of hip-hop and jazz samples' music signals.

The zero-crossing rate is calculated for the hip-hop and jazz genres' signals to understand the waveforms of these signals. For example, the total number of zero-crossing rates for the hip-hop genre is two for ten columns because, two times, the hip-hop signal changes from a positive value to zero to a negative value or from a negative value to zero to a positive value; see Figure 6. This feature is very important for research in the field of music information retrieval.

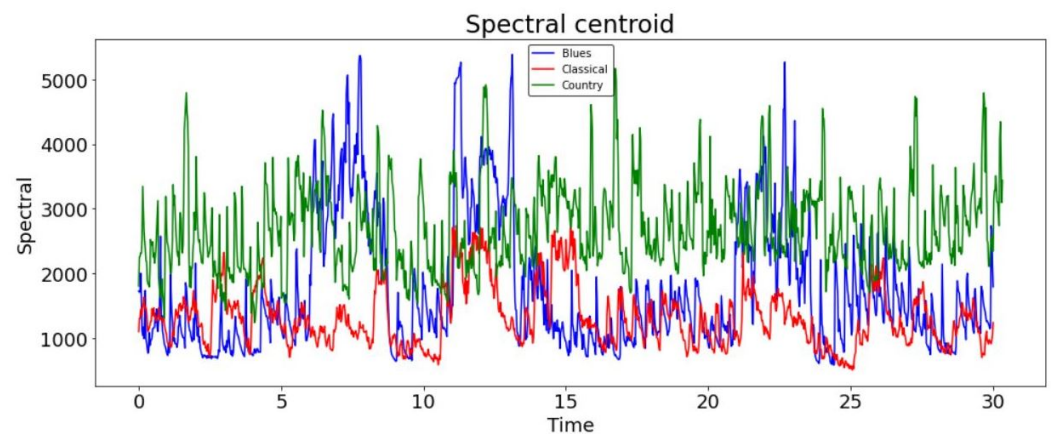


**Figure 6.** Zoomed view of the zero-crossing rate of hip-hop music signal for ten columns.

(b) Root mean square energy—The root mean square demonstrates the loudness of a sound.

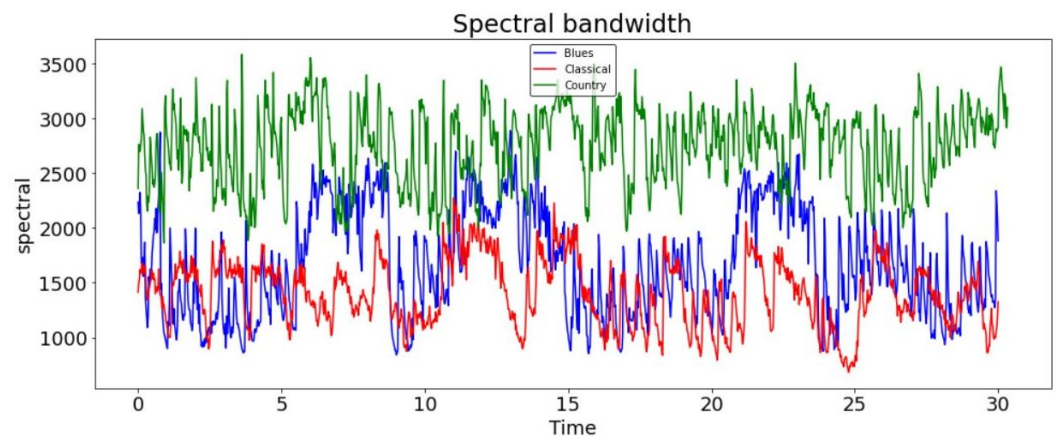
Frequency domain features:

(a) Spectral centroid—The spectral centroid is a measure of the 'centre of mass' for a sound. It is represented by the weighted mean of the frequencies present in the sound. A spectral centroid is a good evaluator of the brightness of an audio signal. A spectral centroid demonstrates the spectral shape (frequency) of an audio signal. Measurements show that the blues signal has a spectral centroid more towards the middle. On the other hand, the country signal has a spectral centroid more towards the end; see Figure 7.



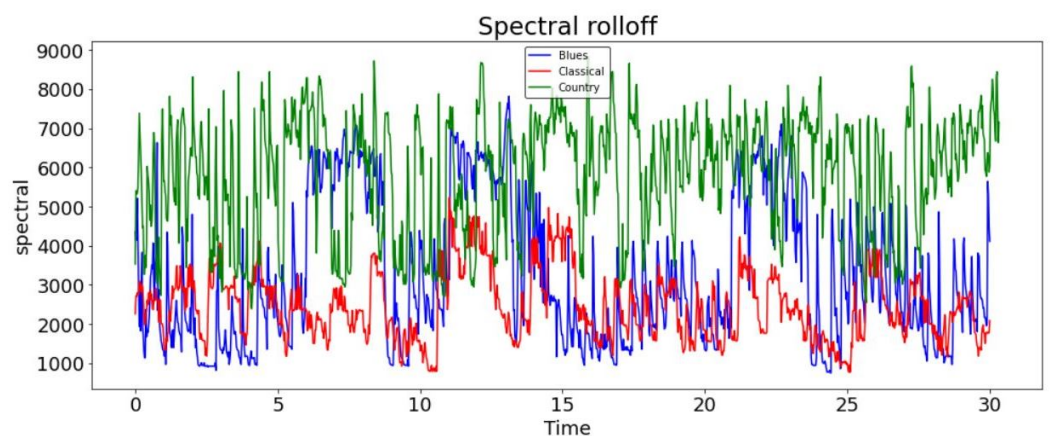
**Figure 7.** Spectral centroid (graphical representation) of blues, classical, and country genres.

(b) Spectral bandwidth—This is represented as the calculated variance from the spectral centroid; see Figure 8.



**Figure 8.** Spectral bandwidth (graphical representation) of blues, classical, and country genres.

(c) Spectral roll-off—The roll-off frequency for each frame in the music signal is calculated by spectral roll-off. The following numerical measurements show the spectral roll-off frequency of the blues, classical, and country music genres. Therefore, spectral centroid, spectral roll-off, and spectral bandwidth are computed and visualised in this paper for the blues, classical, and country genres to analyse the frequency of these audio signals; see Figure 9.



**Figure 9.** Spectral roll-off (graphical representation) of blues, classical, and country genres.

(d) Mel frequency cepstral coefficients—This feature is widely used in speech recognition and audio similarity measurement. In this paper, the Mel frequency cepstral coefficients are a set of features (1 to 20) to define the shape of a music signal for the processing of a music signal. In this process, the music signal is divided into small frames to compute a discrete cosine transform.

### 3.2.2. Data Preprocessing and Exploratory Data Analysis (EDA)

The machine learning algorithm requires a few steps to clean the dataset before training a classifier. The steps are known as data preprocessing. Exploratory data analysis involves the steps to analyse the quality, nature, data type, missing values, and problems in GTZAN data. At first, unwanted entries such as missing or infinite values are dropped to process with a clean dataset and to achieve accurate accuracy for the random forest classifier. The duplicate values are omitted to train and test the machine learning model with unique samples of the dataset. In addition, the count of each class is inspected to avoid the problem of class imbalance. Class imbalance in machine learning defines a problem where a few classes have more samples than other classes. The oversampling or downsampling of the dominant class is the solution to the class imbalance problem. However, in the GTZAN dataset, differences between the class samples of the majority and the minority classes are minimum. Therefore, the oversampling or downsampling solutions are ignored while preparing the dataset for the classification of music genres. However, the performance of the classifier is decreased due to huge differences in the range of music features in the GTZAN dataset. To solve this problem, the normalization technique is used in this paper to scale the music features before training the random forest classifier. Spark supports *StandardScaler* which transforms the vector rows of the dataset. This transformer normalizes each music feature to have a unit standard deviation or zero mean with two parameters, *withStd* and *withMean*. After scaling the music features, the performance accuracy of the random forest classifier is noticeably improved.

Moreover, there are sixty numerical features (dimensionalities) of songs in the GTZAN dataset, such as length, the mean and variance values of the chroma short-term Fourier transform, root mean square, spectral bandwidth, spectral roll-off, spectral centroid, zero-crossing rate, harmonics, perceptual, tempo BPM (beats per minute), and Mel frequency cepstral coefficients (MFCCs). The data types of these numerical measurements are experimented with PySpark codes and converted from string into numeric data types because the classification task is performed with all numeric features of the dataset. In addition, the data are studied, and various quantitative measurements of each music class are calculated in a summarized way (descriptive statistics) to represent the quantitative descriptions of music features of each class (label). Descriptive statistics shows the deviation of music features from mean values. Furthermore, a correlation coefficient is calculated for all mean features, such as the mean of the chroma short-term Fourier transform, root mean square, spectral roll-off, spectral centroid, spectral bandwidth, zero-crossing rate, harmonics, perceptual, tempo BPM (beats per minute), and Mel frequency cepstral coefficients (MFCCs), to understand the statistical relationship between two distributions of data and the strength of the relationship of each feature to the target variable. The correlation coefficient has a value between  $-1.0$  and  $1.0$ .

The correlation matrix, Figure 10, demonstrates the correlation between the means of music features. The matrix represents high positive correlations (value is 1.0) between the same means of features which are highlighted by blue colour on the heat map. This value denotes perfect positive correlations between features. The increase in one variable results in an increase in another variable. The zero correlations (value is 0 or  $-1.0$ ) between different means of music features are highlighted by different colours on the heat map. These variables have a perfect negative relationship between them. Figure 10 also shows that spectral bandwidth mean, spectral centroid mean, spectral roll-off mean, and zero-crossing rate mean have strong positive correlations, which are also highlighted in blue



colour on the heat map. Spectral features are extracted to examine the correlations between these features and music classes (target variable).

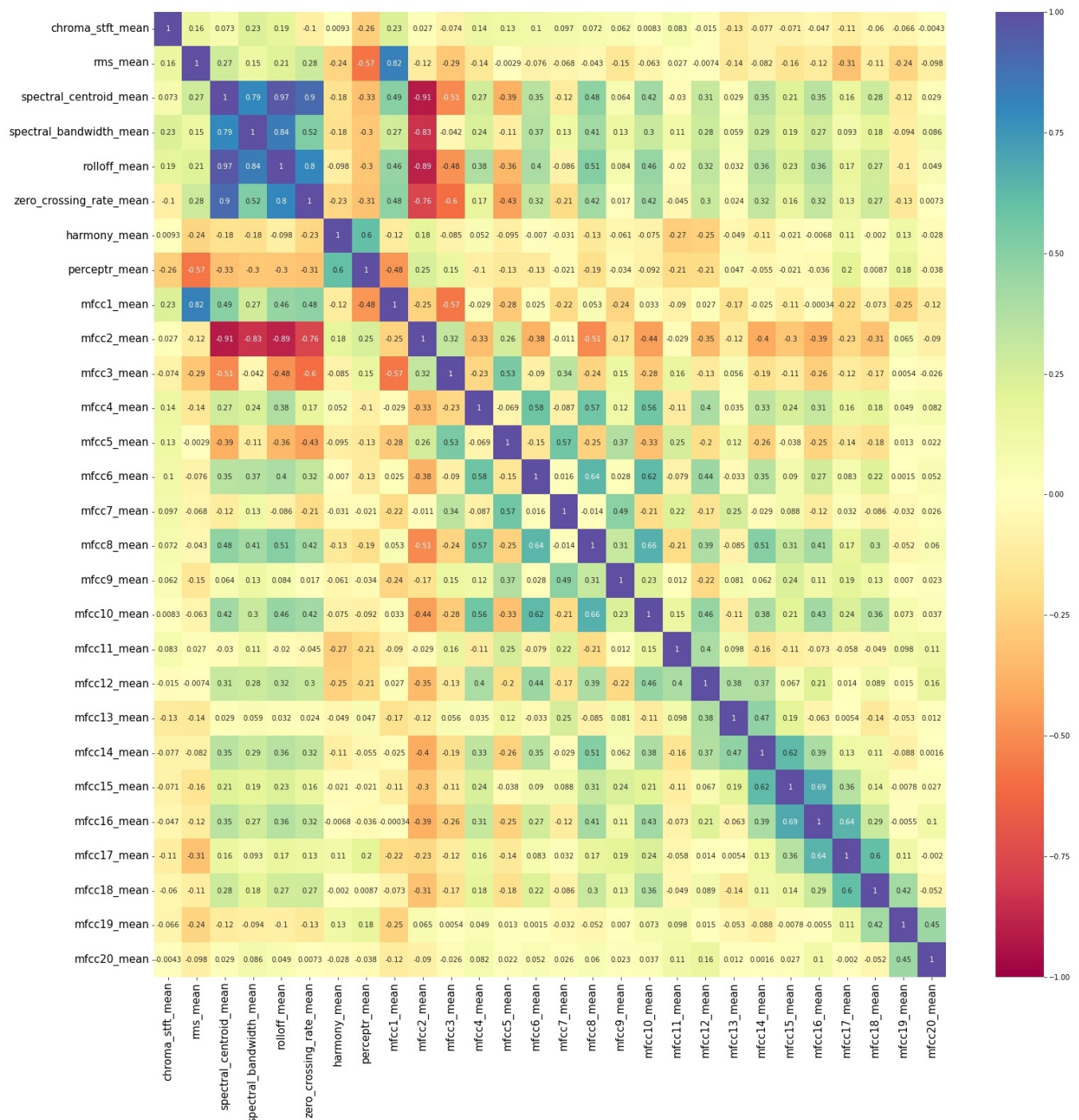


Figure 10. Correlation between means of music features.

Figure 11 indicates the correlations between spectral centroid variance, spectral centroid mean, spectral bandwidth mean, spectral bandwidth variance, spectral roll-off mean, and spectral roll-off variance with the target variable music classes. The diagonal cells denote perfect positive correlations between the same music features which are highlighted in blue colour on the heat map. Figure 11 also depicts that there are very low or zero correlations between spectral features and the target variable music classes which are highlighted in yellow colour on the heat map. This means that the change in spectral



features does not affect the target variable music classes. Therefore, the spectral features of music are less important for giving a higher rate of performance accuracy for the machine learning classifier. Thus, correlation analysis emphasizes the linear relationship between two variables, or independent and target variables. It is noticed that most of the music features of the GTZAN dataset have zero correlation with the target class. Therefore, most of the music features are studied and utilized for the training of the random forest classifier in this paper to achieve a higher rate of performance accuracy which is discussed in the next section, Section 3.3. This is an interesting investigation which is highlighted in this paper.

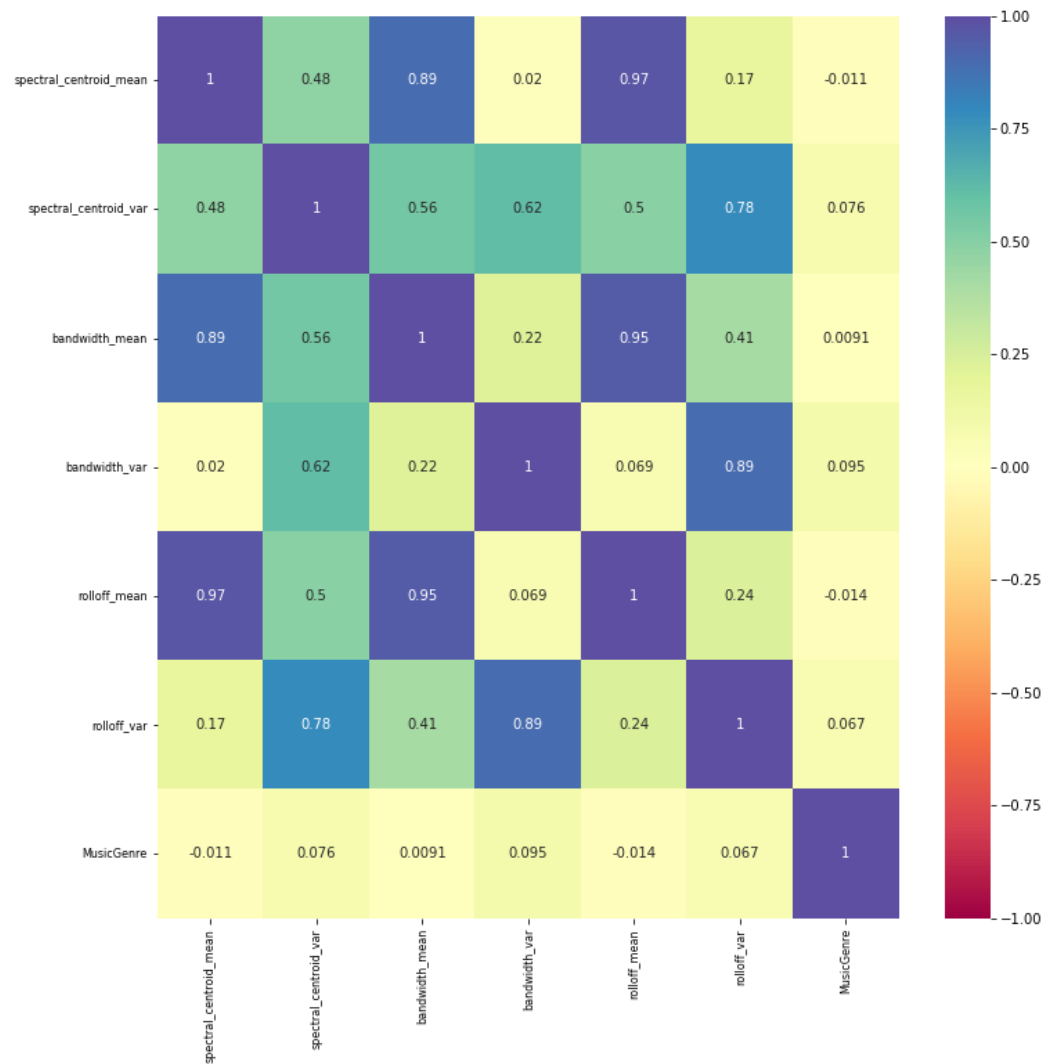


Figure 11. Correlation matrix for spectral features (independent variable) and label (target variable).

### 3.3. Feature Selection

Feature selection is an important aspect of the music genre classification process apart from using appropriate machine learning algorithms and different hyperparameter optimization techniques. The performance of the machine learning algorithm depends on the selection of suitable features. Moreover, the objective of the feature selection is to select a suitable set of features for music genre classification. The GTZAN dataset contains sixty musical features which can increase the duration of training time. By filtering the irrelevant information, not only the performance of the machine learning model can be increased but also the duration of training time can be decreased. Besides that, few features may contain noise and duplicate samples, which can reduce the accuracy of machine learning models. Features can be filtered and complex features can be removed to overcome the problem of overfitting machine learning models. Furthermore, in the GTZAN dataset, there are few

duplicate entries in some features. Those features are dropped to improve the performance of the classifier and to decrease the duration of training time. The random forest algorithm of Apache Spark's scalable machine learning library and data analysis procedures are used to carry out the feature selection and classification process. The classifier also follows the ensembling learning method to decrease variance among music features and achieve the highest classification accuracy for large-scale datasets. The random forest classifier is trained with random features in iteration following the bagging procedure, and the performance accuracy of the classifier is recorded in each iteration. Most music features with higher-importance scores are selected in this paper based on the achieved highest accuracy of the random forest classifier. The selected features are assembled by using the *VectorAssembler* library of PySpark.

### 3.4. Building and Training Procedures of the Random Forest Classifier

Apache Spark is a fast, open-source, distributed analytical engine for big data processing. A large-scale GTZAN dataset is explored in this paper to classify music genres using the random forest machine learning algorithm supported by Apache Spark. The reason for selecting the Apache Spark framework for the classification procedure is not only to reduce the computation time for large-scale data processing but also to utilize Spark's scalable machine learning algorithms to perform the classification task effectively. After cleaning the data and selecting important features, it is necessary to build and train the machine learning classifier to carry out the classification procedure. The GTZAN dataset has a total of 10,990 music samples which are randomly divided into 80% training samples and 20% testing samples. The testing samples are reserved for testing the performance of the machine learning classifier after the training process. The main aim of the training process is to minimize the cost function and bias, which decrease the error between the original value and the predicted value. After splitting music samples, the random forest classifier is built with two initial hyperparameters, *numTrees* and *maxDepth*, following the algorithm supported by Apache Spark's MLlib library to process with multiclass classification. The random forest classifier is chosen in this paper, as this classifier is very effective in classifying both categorical and continuous values. The usage of this classifier can be extended not only to binary classification settings but also to multiclass classification settings. The random forest classifier is an ensemble of multiple decision trees in the present work. In addition, the algorithm trains a set of decision trees in parallel. Each node of a decision tree represents a music feature, each branch denotes a decision rule, and each leaf represents the categorical music class. The decision tree classifiers in the random forest give rise to problems such as overfitting and high variance in the testing samples. However, the random forest classifier combines many decision trees for reducing the risk of overfitting, high variance on the testing samples, and correlations, as well as for estimating the missing samples. Therefore, combining all predictions from each decision tree and voting for the most occurring prediction improves the performance of the classifier on testing samples. Therefore, the predictions from each tree are collected and the classification is based on the categorical class that receives the most votes and more similarity between the original value and the predicted value.

Furthermore, when training a machine learning model with the random forest algorithm, different hyperparameters are applied and optimized. These hyperparameters are supported by the random forest algorithm from the MLlib library of Apache Spark. The hyperparameters are tuned and investigated randomly to improve the performance of the developed random forest classifier. The combination of hyperparameters that gives the best accuracy for the classification of music genre (Section 4.3) is chosen to train the final classifier. The training process of a specific machine learning algorithm is controlled by appropriate hyperparameters supported by the algorithm. For example, *numTrees* and *maxDepth* are hyperparameters that represent the number of trees and the maximum depth of each tree in the random forest. These parameters are optimized in this paper to increase the testing accuracy of the developed random forest classifier. A detailed discussion of the

tuning procedure and a few additional hyperparameters such as *featureSubsetStrategy* and *impurity* are discussed in Section 4.3 of this paper. Increasing another parameter, *maxBins*, allows the algorithm to make split decisions, which is utilized for the hyperparameter tuning process in this paper. The hyperparameter optimization process requires training the random forest machine learning classifier multiple times. After the completion of the training process with optimized hyperparameters, testing data are used to predict and store the outcome in a target variable. The highest-voted predicted class is considered as the final prediction using the random forest classifier. In this paper, the PySpark *Estimator* is used to train the model and the *Transformer* is used to predict the music class.

### 3.5. Verification of the Machine Learning Model's Accuracy

This optional step is used to validate the performance accuracy of the random forest classifier. A decision tree classifier tends to learn the noise in data. This algorithm generally overfits the data, which gives rise to data problems with low bias but high variance. Pruning is a technique that is used in the decision tree to reduce the problem of overfitting. However, the random forest classifier gives good accuracy in classification or regression. This algorithm follows an ensemble learning method which gives rise to many learners and aggregates to one outcome. In this method, the ensemble outcome produces lower bias and lower variance, due to low correlations between individual trees. The training set is iterated by randomly selected features from the actual training set during the training process. This process is called bootstrapping. Therefore, the validation of music samples is not necessary for the random forest algorithm explored in this paper. If the validation process is required, then 80% of the training data can be divided into training and validation samples randomly. The performance accuracy of the machine learning model can be computed using a method called K-fold cross-validation, where the model can be validated through K iterations to measure the training accuracy.

### 3.6. Evaluation of Model Accuracy

This is the final and most critical step of this paper. After completing the training process, it is essential to test the random forest classifier with the testing samples to calculate the accuracy of the model on the testing samples. In the earlier step, 80% of the samples are used to train the random forest classifier. Now, 20% of the reserved samples are used to test the classifier. In this paper, the PySpark library *MulticlassClassificationEvaluator* is used for the classification of multilabel music genres and the *Evaluator* is utilized to evaluate the accuracy of the model.

This paper aims to evaluate the performance accuracy of the random forest classifier for the classification of music genres. At the same time, different machine learning models are evaluated in terms of performance efficiency for the classification of music genres. The most frequently used performance metrics which are explored in this paper to check the performance accuracy of the machine learning classifiers include the confusion matrix, accuracy, recall, precision, and F1-score.

The accuracy of a confusion matrix is measured by the percentage (%) of correctly classified classes for the music genre testing samples. The music genre classification result is calculated and visualised in this paper by using a confusion matrix. For the confusion matrix, the actual class of the sample is represented by each row, and the predicted class of the sample is denoted by each column. In the above matrix, when the actual class and predicted class show a true outcome, it is considered as a true positive (TP). If the actual class shows a true outcome but the predicted class shows a false outcome, then it is known as a true negative (TN). False positive (FP) and false negative (FN) denote the number of incorrectly classified samples. The accuracy Equation (1) is calculated as the fraction of the number of correct predictions and the total number of predictions by the machine learning model. The recall Equation (2) is the accuracy of the model on the actual positive class. The precision Equation (3) is the measurement of actual positive samples out of predicted

positive samples. The accuracy, recall, precision, and F1-score Equation (4) of the music genre classification metric are calculated in this paper.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

Furthermore, the training time duration of the random forest classifier is also evaluated and recorded in this paper. The training duration of a classifier depends on the total number of training samples and the hardware configurations of the platform, which are solved by necessary steps.

#### 4. Experimental Results and Discussions on the Outperformance of the Classifier

In this section, an outline of the findings is discussed after carrying out the experimental analysis with the music features of the GTZAN dataset and training the random forest classifier. Section 4.1 demonstrates the experimental environment used for the classification process. Section 4.2 discusses the procedures to encode the GTZAN dataset into the experimental environment. Section 4.3 demonstrates the tuning process of hyperparameters, followed by Section 4.4, which highlights the performance measurement of the developed random forest classifier using a confusion matrix, accuracy, recall, precision, and F1-score. Section 4.5 demonstrates the comparison of performance accuracy with other classifiers. Section 4.6 contains a discussion on the outperformance of the developed random forest classifier.

##### 4.1. Experimental Environment

The environment is very important for any type of investigation, as it affects the execution of the implementation. Multiple machine learning classifiers, such as decision tree, naïve Bayes, random forest, and logistic regression, are implemented in this paper using PySpark, which is the integration of Python programming and Apache Spark. These machine learning classifiers follow the algorithm from Apache Spark's MLib library. The entire implementation and execution are completed in the Google Collaboratory environment, which is the Python 3 Google Compute Engine backend. Different Python libraries, such as Matplotlib, Pandas, NumPy, Seaborn, librosa, and IPython, are utilized in this paper for the implementation of music data analysis and classification. Apache Spark, which is an open-source memory-based framework, is used in this paper to reduce the duration of big data processing with no computational cost. In this paper, a single machine running a spark engine is used. The hardware environment utilized in this paper is a laptop powered by an Intel (R) Core (TM) i5-1035G1 Central Processing Unit. The processor is with 8 GB of random access memory. The system operates on Windows 10 personal 64-bit operating system.

##### 4.2. Encoding of the GTZAN Dataset

In this section, the result of investigating the popular GTZAN dataset is discussed. The dataset is accessed and downloaded from a public weblink and loaded in Google Drive. After launching the SparkSession, the dataset is read in the session to develop an application. The dataset has audio files, images, and two CSV files. The dataset contains a total of 10,990 music samples with ten categorical classes. It has duplicate values which are removed by data preprocessing techniques discussed in Section 3.2.2. The audio features are analysed and visualised to understand the nature of the GTZAN dataset. After cleaning

the dataset, the random forest classifier is trained with the training samples and tested with the testing samples. The classifier manages to classify music genres with 90% accuracy on the testing samples.

Moreover, the GTZAN dataset has music excerpts in .wav (waveform) format. There are time and frequency domain music features of the GTZAN dataset discussed in Section 3.2.1. In this paper, several sample songs from each genre are investigated in waveform format to analyse the signals and sample rate of each sample song. An audio signal is the amplitude of the sound of a sample song which takes normalized values from  $-0$  to  $1$ . In addition to that, the values of signal and sample rate for each sample song from the GTZAN dataset are calculated to understand the numerical measurement of the sample song excluding numerical features.

#### 4.3. Tuning of Hyperparameters

Hyperparameter tuning is a process of setting the right hyperparameters to improve the accuracy and precision of machine learning classifiers. Optimizing hyperparameters is the most challenging part of designing a machine learning model. The developed random forest classifier has two primary hyperparameters which are tuned to increase the accuracy of the classifier in this paper. The first hyperparameter is *numTrees* which counts the number of trees in the random forest. The value of this hyperparameter is increased in this paper to increase the number of trees, which reduces the variance in predictions. The low variance increases the test-time accuracy of the model. The second hyperparameter is *maxDepth* which is the maximum depth of each decision tree in the random forest. This hyperparameter is also increased to improve the performance of the random forest classifier. The deep trees make a machine learning model more powerful to achieve greater testing accuracy. A single decision tree always tends to raise the problem of overfitting more than multiple trees. Therefore, it is important to increase the depth of trees to generate higher accuracy. Increasing the number of trees and the maximum depth of each tree in the forest can lead to a decrease in the variance in predictions. The random forest classifier is initially trained with these parameters and the performance is recorded. The values of the initial parameters *numTrees* and *maxDepth* were 20 and 10, respectively. The initial accuracy of the random forest classifier was 62%; see Table 3 in this work.

**Table 3.** Initial hyperparameters of the developed random forest classifier.

Initial Hyperparameter	Value	Accuracy of Random Forest Classifier
<i>numTrees</i>	20	62%
<i>maxDepth</i>	10	62%

In this paper, the initial parameters *numTrees* and *maxDepth* are tuned and more additional hyperparameters are implemented, such as *featureSubsetStrategy*, *impurity*, *subsamplingRate*, and *maxBins*, to achieve the highest classification accuracy of the developed random forest classifier; see Table 4. The *maxBins* denotes the number of bins used to split continuous features and make accurate split decisions. This parameter is effective for increasing the computation of machine learning predictions. The *impurity* defines the effectiveness of trees in splitting the data. The *impurity* is set to *Gini* in this paper to compute the probability of incorrectly classified features that are selected randomly. The *featureSubsetStrategy* is set to *auto* to reduce the subset of features automatically. The *subsamplingRate* denotes the size of the training dataset used for training each tree in the random forest. The initial hyperparameters are tuned with a different set of values to record the best accuracy of the random forest classifier. Table 4 demonstrates that the classifier has managed to achieve 90% accuracy for music genre classification after tuning the hyperparameters.

**Table 4.** Tuned and additional hyperparameters of the developed random forest classifier.

Tuned Hyperparameter	Value	Additional Hyperparameter	Value	Accuracy of the Random Forest Classifier
numTrees	200	subsamplingRate	1.0	90%
maxDepth	23	featureSubsetStrategy	auto	90%
maxBins	132	-	-	90%
impurity	Gini	-	-	90%

#### 4.4. Performance Measurement

In this section, the accuracy of the random forest classifier for the classification of music genre is computed in terms of the confusion matrix, accuracy, recall, precision, and F1-score. The classification of music genres using the random forest classifier is also visualised through a confusion matrix.

##### 4.4.1. Confusion Matrix

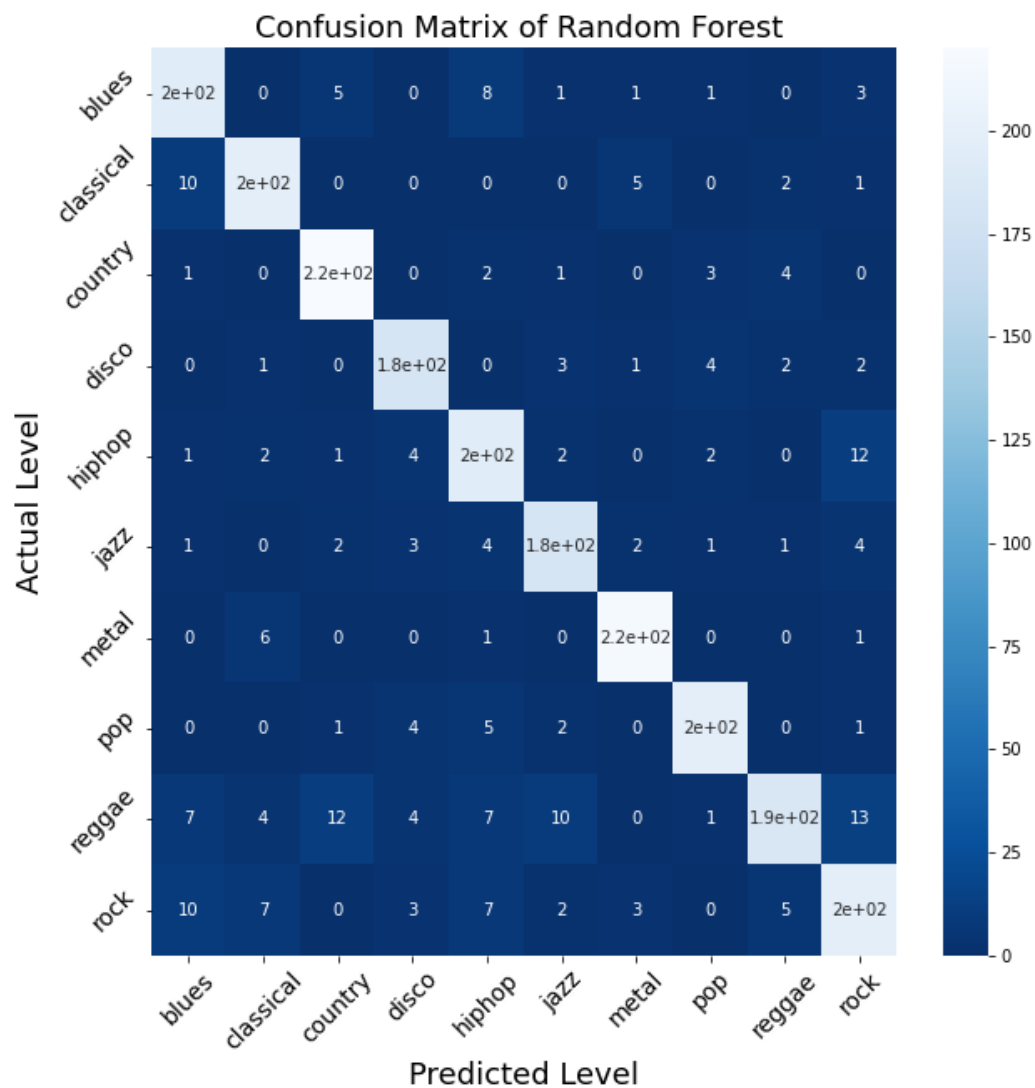
A confusion matrix is a mathematical table which represents the performance of machine learning models on the testing samples. This matrix can be considered as an error matrix. In this paper, the confusion matrix depicts the performance of the random forest classifier on the testing samples with known actual classes Figure 12. The GTZAN dataset has a total of 10,990 music samples. The training samples are 8800 and the testing samples are 2190, respectively. The testing samples are separated from the training samples to generate correct accuracy by the machine learning model. The random forest classifier is tested with 20% of the reserved music samples. The random forest classifier in this paper manages to achieve 90% accuracy for the classification of ten music genres' categorical classes with a 15 min processing duration. The diagonal cells of the confusion matrix represent the correctly classified classes (true positive), and the off-diagonal cells represent the incorrectly classified samples. The columns of the confusion matrix indicate the predicted class (false positive) by the developed random forest classifier. In addition, the rows demonstrate the actual class (false negative) by the classifier.

The confusion matrix for the classification of music genres is analysed to find the correctly and incorrectly classified samples of each class. For example, the class metal has 216 correctly classified samples and 20 incorrectly classified samples. Similarly, the class pop has 198 correctly classified samples and 25 incorrectly classified samples. Overall, 1971 music classes are correctly classified (true positive) out of 2190 testing samples, and 219 music classes are incorrectly classified. The confusion matrix demonstrates the relation between the actual and predicted target classes. The matrix also represents the best performance of each music class, which has a lighter colour on the heat map. These classes have higher true positive and true negative cases. From the visualisation in Figure 12, it is noticed that the classes country and metal have higher true positive and true negative cases than the rest of the classes.

It is essential to examine the correctly and incorrectly classified music samples of each class for testing samples to acknowledge the efficiency of the developed random forest classifier. Table 5 displays correctly and incorrectly classified music samples (testing samples) for each music class.

The classification result enumerates that the classes jazz and disco have the least numbers of correctly classified cases. On the other hand, the classes country and metal have the highest numbers of correctly classified cases. In addition, the total number of false negatives for the blues class is denoted by 19 (the sum of the values in the corresponding row) and the total number of false positives for the blues class is denoted by 30 (the sum of values in the corresponding column).





**Figure 12.** Visualisation of music genre classification results (testing samples).

**Table 5.** Correctly (true positive) and incorrectly (false negative or false positive) classified music samples (testing samples) for each music class by the developed random forest classifier.

Music Class	True Positive	False Negative	False Positive	True Negative
Blues	195	19	30	1946
Classical	197	18	20	1955
Country	220	11	21	1938
Disco	183	13	17	1976
Hip-hop	195	24	34	1937
Jazz	183	18	21	1968
Metal	216	8	12	1954
Pop	198	13	12	1967
Reggae	186	58	14	1932
Rock	198	37	37	1918

#### 4.4.2. The Classification Report in Terms of Accuracy, Recall, Precision, and F1-Score

The classification report in this paper, Table 6, enumerates the accuracy, recall, precision, and F1-score for ten music classes. Accuracy shows the performance accuracy of the random forest classifier, which is 90%. The precision value denotes that 90% of the music classes are originally positive out of the predicted positive classes, which represents how many selected classes are actual. Recall shows 91% of the actual positive class that the random forest classifier has identified, which represents how many actual classes are selected by the classifier. The F1-score is denoted as the weighted average of the precision and recall values, which is 91% for the classification of music genre.

**Table 6.** The classification report for ten music genres.

Accuracy	F1-Score	Precision	Recall
0.9	0.912037037037037	0.9078341013824884	0.9162790697674419

#### 4.5. Comparative Analysis with Other Classifiers

Apache Spark supports multiple classification algorithms to solve a multiclassification problem for continuous or discrete data. In this paper, multiple classifiers from Apache Spark's MLlib library are investigated on the GTZAN dataset for ten music genres' classification. The implemented classifiers are logistic regression, decision tree, and naïve Bayes. However, the experimental result shows that the classification rate of the random forest classifier is higher than that of the other classifiers on the GTZAN dataset. The reason is that the random forest forms independent decision tree classifiers on bootstrapped samples of the dataset and then selects the best prediction among all the predictions generated by multiple independent decision trees. This process is defined as an ensemble learning method.

The confusion matrices, accuracies, recall values, precision values, and F1-scores of these classifiers are also compared for the classification of music genres. The random forest classifier outperforms the other classifiers in all measures of classification, followed by logistic regression. The naïve Bayes classifier has a worse performance than other classifiers in all measures. The recall measurement presents that the random forest classifier has more ability to predict the majority of actual music samples than other classifiers, followed by logistic regression. The precision measurement shows how many selected music samples are actual. The random forest classifier also outperforms other classifiers in precision measurement. Therefore, Table 7 highlights the outperformance of the classifier in all classification measures for the classification of music genres.

**Table 7.** Performance of multiple machine learning classifiers on the GTZAN dataset.

Classifier	Accuracy	F1-Score	Precision	Recall
Random forest	0.9	0.9120	0.9078	0.9162
Decision tree	0.6191	0.6744	0.6744	0.6744
Naïve Bayes	0.4281	0.3143	0.2875	0.3467
Logistic regression	0.7276	0.7980	0.7567	0.8442

#### 4.6. Discussion on the Outperformance of the Random Forest Classifier

This subsection encapsulates all unique characteristics of Apache Spark's ensemble learning classifier random forest designed in this paper for music genre classification.

##### 4.6.1. Effectiveness

Apache Spark supports an ensemble learning algorithm random forest, which designs a machine learning classifier composed of a set of other base models. Random forest uses decision trees as its base machine learning classifiers. It is a supervised learning

method that works efficiently on categorical, continuous multiclass variables. Decision trees are highly sensitive to training data which could lead to a high variance problem. As a result, a decision tree classifier might fail to predict the correct outcome. The random forest classifier is effective for overcoming this problem. This machine learning classifier combines many decision tree classifiers to create a random forest. In other words, the independent classifiers are combined to form a strong classifier to achieve the best accuracy. Moreover, the GTZAN dataset in this paper is randomly divided into many subsets with random sampling with a replacement procedure. This process of creating a new dataset from the original dataset is called the bootstrapping procedure. At each split of a decision tree, these randomly selected subsets (bootstrapped) are used to train the music data in parallel and to cast the classification votes on a class from each node. After that, the random forest classifier aggregates all the classification votes into a single prediction which is considered the best prediction. Many decision trees of music features are combined to form a random forest to reduce the risk of model overfitting and high variance in the testing dataset. This is another effective characteristic of the random forest classifier in this work. The uncorrelatedness of the predictions of individual decision tree classifiers is the cause of the best effectiveness of the random forest classifier chosen in this paper. The process of combining the bootstrapping and aggregating procedures is called bagging. Instead of selecting the most important features for training, random forest selects the best feature among a random subset of features. Besides that, this classifier can handle a massive amount of data in the classification and regression process effectively. The random forest classifier is chosen in this paper because it is computationally effective in producing more accurate predictions. The classifier is not effective in producing good accuracy while training with nonlinear features.

#### 4.6.2. Robustness

Robustness defines the strength of the random forest classifier developed in this paper. The classifier in the present work can work well with categorical features and multiclass classification. This classifier does not require any feature scaling because the random method of selecting features decreases the correlation between features. This reduces the noise in data and removes the outliers. The random forest classifier can handle any missing feature values in a dataset. This classifier creates independent decision trees on music data samples. These decision trees are trained independently using bootstrapped random samples of data. This procedure helps the classifier to be more robust than an individual decision tree for reducing the risk of overfitting problems. Moreover, prediction results are being received from each decision tree and the best solution is selected by the classifier. The higher number of trees in the forest generates higher accuracy in classification results. This feature also makes a random forest classifier robust, which is highlighted in the paper by tuning the hyperparameter *numTrees* in *PySpark*. The random forest is highly robust to the noise of data by reducing outliers and tuning hyperparameters. Besides that, if the classification result of one decision tree in a random forest fails due to a weak learning feature, the same classification result produced by other decision trees in the same random forest increases the strength of the final classification result. This unique characteristic makes a random forest classifier more robust towards achieving the best classification outcome.

#### 4.6.3. Convergence

The convergence defines the limitation of a process when evaluating the best performance of a machine learning model. The random forest algorithm emphasizes an iterative process while computing machine learning predictions. The iteration process goes on until it reaches the final stable-point solution of a classification or regression problem. This behaviour of an iterative algorithm is defined as convergence. In this paper, the random forest classifier is trained iteratively with a bootstrapped dataset to classify music genres. Moreover, hyperparameters are optimized in an iterative way to report a set of solutions

in each iteration and to select the best solution. The termination of the hyperparameter optimization process to achieve the best classification accuracy is known as the convergence stage of the random forest classifier in the present work. When the random forest classifier reaches the convergence stage, the further hyperparameter optimization process is not effective for the classification in the present work. The convergence analysis demonstrates that a machine learning model converges when the loss decreases towards its minimum value and reaches its minimum value.

#### 4.6.4. Tuning Hyperparameters

Hyperparameter tuning or optimization is a process of choosing a set of right parameters to achieve the best performance accuracy of a machine learning model. It is observed that the developed random forest classifier achieves 90% performance accuracy on the testing music samples after adding and tuning a set of unique hyperparameters. The hyperparameters are *numTrees*, *maxDepth*, *maxBins*, *impurity*, *featureSubsetStrategy*, and *subsamplingRate*, supported by Apache Spark. The *numTrees*, *maxDepth*, *maxBins*, and *subsamplingRate* hyperparameters are defined with numeric values and *impurity* and *featureSubsetStrategy* are defined with string values. The hyperparameter optimization method is applied to the random forest model while training the model to outperform the model's accuracy in the present work. The *numTrees* hyperparameter, which denotes the number of trees in the random forest, is the basic hyperparameter of this classifier. More hyperparameters, such as *maxDepth*, *maxBins*, *impurity*, *featureSubsetStrategy*, and *subsamplingRate*, are investigated on this classifier to outperform the random forest model in the present work. The maximum depth of each tree and the number of trees in the forest are optimized in this work, as increasing the number of trees and the depth of each tree reduces the variance in predictions, making the random forest classifier more accurate and powerful.

#### 4.6.5. Accuracy

Random forest is considered the best machine learning model for the multiclassification task. It contains multiple decision trees. These decision trees are trained with a random subset of training data and generate output. Random forest aggregates all outcomes and generates the best result according to the maximum voting of decision trees. Thus, random forest generates better classification accuracy than decision trees. In the present work, the random forest classifier achieves 90 % accuracy for a testing sample of music data. This is the best accuracy of the random forest classifier supported by Apache Spark on the GTZAN dataset compared to other works in the literature. Research by Sturm (2013) [41] revealed that mislabelling, repetitions, and distortions are faults of the GTZAN dataset, which can be a challenge for accurate results derived from it. To overcome this challenge, the random forest classifier is utilized for the classification of music genres in the present work. However, random search cross-validation and grid search cross-validation methods using Scikit-Learn tools supported by Python programming can be used for hyperparameter tuning to increase the accuracy of the random forest classifier. Apache Spark does not support these two methods. Thus, the random forest classifier in the present work achieves the best accuracy of 90 % after optimizing a set of unique hyperparameters supported by Apache Spark.

## 5. Analysis and Discussion

The purpose of this quantitative research is to analyse music features and to recognize music genres based on investigated music features. This section highlights a discussion of key findings of the present work for the classification of music genres, an analysis of audio data, a hybrid approach to recognizing music genres, extracting music information from a dataset and classifying the kind of music based on the extracted music information, and the evaluation of different machine learning models for recognizing music genres. A comparison of critical evaluations and the limitations of this research with relevant pieces

in the literature is also included in this section. The present research is the solution of three research questions outlined below.

- How can an audio dataset be analysed and classified into a group of similar kinds of audio?
- What is the best possible way to achieve the highest rate of classification accuracy?
- Which technology can be used to reduce the duration of data processing without computational cost?

The present research demonstrates the analysis of a benchmark music dataset and the prediction of music genres from the dataset that answers the first research questions. The GTZAN dataset has been widely used in the domain of audio feature analysis and music genre classification over the last decade. The time and frequency domain music features, such as spectral bandwidth, spectral centroid, spectral roll-off, and zero-crossing rates, are analysed and visualised for randomly selected blues, classical, and country music samples to understand the frequency of waveforms and the quality of those music samples. This is one interesting finding in the domain of audio data analysis. The frequency signals, sample rate, and length of sample music data are computed to encode and realize sample music signals for ten genres. The exploratory data analysis (EDA) confirms the relations and distributions of music data. There are sixty statistical music features available in the GTZAN dataset. There is a low statistical correlation among different mean music features. In addition, a correlation coefficient matrix is calculated for a few extracted features with the target variable. The result of the computing correlation coefficient demonstrates a low correlation between music features and target variable genres, which indicates that the distance between the variance and mean values of a few music features is high.

After preprocessing and splitting data into training and testing samples, multiple machine learning classifiers are trained and tested to classify music genres into ten different classes. Among these classifiers, the random forest classifier achieves the highest accuracy rate of 0.9 for recognizing the kinds of similar music samples from the testing dataset. The complete dataset is split into 8800 training samples and 2190 testing samples. The classification result indicates that 1971 music classes are correctly classified out of 2190 testing samples, and 219 music classes are incorrectly classified. The recall, precision, and F1-score of the designed random forest classifier are above 90%.

Random forest is a combination of many decision trees to reduce the risk of overfitting and noise in the dataset. It uses the ensemble learning method and follows a supervised learning algorithm. An ensemble learning method uses independent classifiers. This independent classifier either uses different algorithms on the same training samples or uses a similar algorithm trained on different subsets of the training sample. This classifier follows the bagging procedure to generate the classification result. In the bagging procedure, the dataset is randomly divided into different subsamples or training samples to train the same algorithm in parallel. After that, the individual predictions of those classifiers are combined to generate a final prediction. The combining procedure is conducted either by voting or averaging the individual predictions. Therefore, a set of decision trees in a random forest are trained in parallel by randomly selected features. The randomly selected features are generated with a replacement procedure from the original training samples. These features help the random forest model to reduce correlations between the feature attributes. The votes from each decision tree are collected and aggregated to a single class following the bagging procedure. The classification result depends on the selected class which receives the most votes, which answers the second research question. An in-memory, distributed, open-source, cluster computing framework Apache Spark is used to reduce the processing time of machine learning predictions with no computational cost, which answers the third research question. This paper contains the appropriate data and methodologies for solving classification problems in the domain of music. This can be considered as the soundness and scientific impact of this paper.

As mentioned in the literature review, several machine learning algorithms classify music genres for the recognition of kinds of songs. Several datasets are used for this research. An article by Cai and Zhang (2022) [42] reviewed four datasets, GTZAN, GTZAN-NEW, ISMIR2004, and Homburg, and extracted the spectral and acoustic features of music with an auditory image to improve the classification accuracy of music genres. In this research, a single dataset GTZAN was used to analyse for the classification of music features. All sixty features were investigated for the classification procedure to improve the accuracy of the machine learning model. Tzanetakis and Cook (2002) [4] focused on several feature extraction procedures to solve the classification problem. Different features, such as rhythmic content features, timbral texture features, and pitch content features, were analysed for automatic music genre classification with a 61% accuracy rate. The methodological choices are constrained by many researchers. Several music features, such as zero-crossing rate, chroma short-term Fourier transform, root mean square, spectral centroid, tempo BPM (beats per minute), spectral bandwidth, spectral roll-off, harmonics, perceptual, and Mel frequency cepstral coefficients (MFCCs), are analysed to understand the characteristics of music signals in many research areas. The short-term Fourier transform is calculated to convert a sound signal from a time domain to a frequency domain and its spectrum is visualised to acknowledge the intervals of frequencies in music signals. Similarly, time and frequency domain features are analysed in the present work to assess the nature of music or sound signals in different segments of time or frequency. However, the statistical relationships of music features are also computed through a correlation coefficient matrix and exploratory data analysis (EDA). Section 3.2.2 is used to acknowledge the distributions of data before choosing an appropriate machine learning algorithm in the present work. This result, which is an interesting key finding of this research, has not previously been described in any pieces in the literature. The exploratory data analysis method is also applied to the features of each categorical label, blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock to filter the data distributions of each music category. The present research is designed to determine the effect of implementing only statistical features to an appropriate machine learning algorithm to generate the classification result. The histograms of the Daubechies wavelet coefficients (Daubechies wavelet coefficient histograms) of music signals were computed in a comparative study [43] to capture the local and global information of music signals. This new feature improved the accuracy of music genre classification. However, the correlation coefficient and descriptive statistics of statistical features are computed in this paper to measure the relations between different features and the target variable. Karunakaran and Nagamanoj (2018) [44] investigated a hybrid classifier on the GTZAN dataset and Free Music Archive (FMA) Dataset. An audio feature extraction tool Essentia and a neural network were implemented on the datasets. However, an ensemble learning approach with a distributed framework Apache Spark is adopted in the present work for the classification of music genres. Furthermore, after understanding the distributions of data, it is necessary to divide the dataset into training, validation, and testing samples. Several pieces of literature have highlighted the splitting of a dataset into training and testing samples. Researchers have used the K-fold cross-validation method to break the symmetry of data and to estimate the efficiency of a machine learning model on unused data. This validation results in low bias in machine learning models. Elbir, İlhan, Serbes, and Aydın (2018) [45] proposed a K-validation method to validate some portion of training data to test the support vector machine and the random forest classifier for music genre classification. However, the bagging methodology is used in the present research to distribute the music dataset by resampling. The weak learners are trained using resampled training sets to generate a voting process of weight parameters. A random forest classifier combines all votes from weak learners to generate a final strong decision. This is another important finding of this research paper. The GTZAN dataset is divided into 80% training and 20% testing samples in this paper to classify music genres. Alternatively, a series of machine learning classifiers, including convolutional neural network, artificial neural network, support vector machine, naïve Bayes, K-nearest neighbor, decision tree,



recurrent neural network, logistic regression, and random forest, have been investigated and implemented for recognizing the kind of music in recent pieces in the literature. In this paper, the investigation is carried out with decision tree, naïve Bayes, random forest, and logistic regression classifiers for music genre classification. While previous research has focused on deep learning models, traditional neural networks, and multiple machine learning algorithms, the present research focuses on the implementation of the various machine learning classifiers from Apache Spark’s scalable machine learning library for recognizing the types of music. The present experiment provides new insight into the efficiency of Apache Spark and its scalable machine learning library in the processing of big-size datasets for the classification of music genres. Mayer and Rauber (2011) [46] stated that the combination of large datasets and best-performing feature sets increases the performance accuracy of machine learning models. However, there is an inconsistency with this statement. The duration of the machine learning model’s training and prediction computation increases while processing large datasets. The investigation would have been more useful if it had used the distributed computing framework Apache Spark to process large-scale datasets for music genre classification. In the present work, Apache Spark is used for the in-memory faster computation of machine learning predictions to recognize the types of music for a large-volume GTZAN dataset. The main strength of running machine learning applications in Apache Spark is the reduction in the duration of machine learning prediction computation from minutes or hours to seconds in the present work; see Table 8. This is the most important finding of the present research.

**Table 8.** Training and prediction durations of machine learning classifiers in the present work.

Machine Learning Classifier	Accuracy (Test Set)	Training Duration	Prediction Duration
Random forest	0.9	15 min	10 s
Logistic regression	0.72	30 s	4 s
Decision Tree	0.62	12 s	5 s
Naïve Bayes	0.42	8 s	4 s

Another paper by Devaki and Sivanandan (2021) [47] emphasized the application of multiple machine learning algorithms on the GTZAN for music genre classification. The random forest classifier provided the best performance among the other machine learning classifiers with an accuracy of 70%. The analysis relied too heavily on content-based and frequency domain features. However, all statistical features of the GTZAN dataset are analysed and scaled for the classification of music genres in this paper. The hyperparameters of the developed random forest classifier are tuned to improve the performance accuracy of the classifier. The tuned random forest manages to achieve 90% accuracy in the present work with a 15 min processing duration. This is the major implication of the present research, which contradicts the earlier contributions of researchers.

## 6. Conclusions

This research paper reveals the solution of reducing the computation duration of machine learning predictions without computational cost for a large-scale dataset using Apache Spark. This paper also presents the analysis of several music features and the development of a machine learning classifier to classify music genre. The GTZAN dataset is identified for the exploration of music features and classifying them into groups of similar music. Multiple machine learning classifiers supported by Apache Spark, including decision tree, random forest, naïve Bayes, and logistic regression, are implemented on the GTZAN dataset for the classification of music genres. The workflow of an ensemble learning algorithm is introduced in the domain of music genre classification. Time and frequency domain music features are analysed to acknowledge the signals of music data. Moreover, the statistical features of music data are computed to understand the distribution

of audio (music) data. The correlation coefficient is calculated to assess the relation between several music features and the target class. From the investigation, it is concluded that the random forest classifier is the best classifier for better performance in the music genre classification task due to its voting process of classification. The logistic regression and decision tree classifiers are suitable for short-term performances in any organization where music classes are less targeted.

A study [41] said that the GTZAN dataset has mislabelling, repetition, and distortion problems, which could reduce the performance of the machine learning classifier. The investigation in the present work also indicates that the proposed random forest classifier can manage noise in data, missing values in data, and overfitting problems. A few data preprocessing steps are performed to study all features of the dataset before splitting the dataset into training and testing samples. After training the random forest classifier with the randomized statistical features of the GTZAN dataset, the classifier produces a test accuracy of 90% for the classification of music genres. This is the best accuracy achieved by the random forest classifier for the GTZAN dataset in recent years, which is the primary contribution of the present work. Besides that, a piece of massive information on music features, correlation analysis, and statistical measures for the GTZAN dataset is provided in this work.

Furthermore, an in-memory, distributed computing, parallel processing, fault-tolerant framework Apache Spark is used to process the big-size GTZAN dataset. This framework reduces computation time for machine learning predictions, as it processes data parallelly and provides no cost for computation. This is an interesting contribution of this paper, which is highlighted to fill the gap in the literature in recent years. The combination of Apache Spark and its scalable machine learning libraries in the domain of music genre classification is highlighted in the present work. The classification result is visualised by using a confusion matrix.

## 7. Future Insights

In the future, the developed random forest classifier can be investigated on the second music dataset to evaluate the performance of the classifier. At the same time, the tuned random forest can be trained by one dataset and tested by another dataset to evaluate the classification accuracy in the domain of music genre classification. Moreover, the multiclass classification can be converted to binary classification and the random forest classifier can be modified to achieve the requirement. The investigation can be more complex if the developed random forest classifier is implemented on multiple music datasets to evaluate the performance of the classifier. A comparative analysis can be visualised to estimate the performance of the developed random forest classifier on multiple datasets. Alternatively, other machine learning approaches, such as support vector machine, artificial neural network, K-nearest neighbor, recurrent neural network, and convolutional neural network (deep learning), can also be applied to the GTZAN dataset for the classification of music genres. However, the training duration and computational cost of machine learning predictions increase with these machine learning classifiers for large-scale datasets. This could lead the way in developing a problematic situation in any organization where fast, scalable data processing is required to predict a massive range of music genres in a real-world scenario. Therefore, the implementation of this random forest classifier supported by Apache Spark for predicting music genres is crucial to working with massive amounts of data in a real-world scenario. The number of features for each music genre can be increased to improve the performance accuracy of any classifier.

For extensive further work, it is recommended to create a hybrid classifier by integrating two or more machine learning algorithms and implementing them on the same dataset to examine the performance of the hybrid classifier. The present work also recommends that a fault-tolerant distributed computing framework Apache Spark should be considered for processing large-scale datasets. Besides that, the developed backend Apache Spark engine can be executed as a single Python file to run a music recognition application. In addition,

several applications, such as music similarity analysis, music sentiment analysis, music recommendation, artist identification, and instrument identification, can be developed by using different algorithms of Apache Spark's scalable machine learning library. The Amazon SageMaker service of Amazon Web Services (AWS) can be used to build, train, and deploy the machine learning model. This can be interesting future work in the domain of machine learning applications.

**Author Contributions:** Conceptualization, M.C. and A.K.; methodology, M.C.; software, M.C.; validation, M.C., A.K. and M.A.G.; formal analysis, M.C.; investigation, M.C., A.K. and M.A.G.; resources, M.C.; data curation, M.C.; writing—original draft preparation, M.C. and A.K.; writing—review and editing, A.K. and M.A.G.; visualization, M.C.; supervision, A.K. and M.A.G.; project administration, M.C. and A.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The dataset, environment setup and machine learning used during the current study are available from the corresponding author on reasonable request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Wu, J.; Hong, Q.; Cao, M.; Liu, Y.; Fujita, H. A group consensus-based travel destination evaluation method with online reviews. *Appl. Intell.* **2022**, *52*, 1306–1324. [[CrossRef](#)]
2. Zhao, C.; Chang, X.; Xie, T.; Fujita, H.; Wu, J. Unsupervised anomaly detection based method of risk evaluation for road traffic accident. *Appl. Intell.* **2022**, 1–16. [[CrossRef](#)]
3. Ganeva, M.G. Music Digitalization and Its Effects on the Finnish Music Industry Stakeholders. Ph.D. Thesis, Turku School of Economics, Turku, Finland, June 2012.
4. Tzanetakis, G.; Cook, P. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Proc.* **2002**, *10*, 293–302. [[CrossRef](#)]
5. Chen, K.; Gao, S.; Zhu, Y.; Sun, Q. Music genres classification using text categorization method. In Proceedings of the 2006 IEEE Workshop on Multimedia Signal Processing, Victoria, BC, Canada, 3–6 October 2006; pp. 221–224.
6. Dai, J.; Liang, S.; Xue, W.; Ni, C.; Liu, W. Long short-term memory recurrent neural network based segment features for music genre classification. In Proceedings of the 2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP), Tianjin, China, 17–20 October 2016; pp. 1–5.
7. Sanden, C.; Zhang, J.Z. Enhancing multi-label music genre classification through ensemble techniques. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 24–28 July 2011; pp. 705–714.
8. Vishnupriya, S.; Meenakshi, K. Automatic music genre classification using convolution neural network. In Proceedings of the 2018 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 4–6 January 2018; pp. 1–4.
9. Ajoodha, R.; Klein, R.; Rosman, B. Single-labelled music genre classification using content-based features. In Proceedings of the 2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), Port Elizabeth, South Africa, 26–27 November 2015; pp. 66–71.
10. Bahuleyan, H. Music genre classification using machine learning techniques. *arXiv* **2018**, arXiv:1804.01149.
11. Silla, C.N.; Koerich, A.L.; Kaestner, C.A. A machine learning approach to automatic music genre classification. *J. Braz. Comput. Soc.* **2008**, *14*, 7–18. [[CrossRef](#)]
12. Karami, A.; Guerrero-Zapata, M. A fuzzy anomaly detection system based on hybrid PSO-Kmeans algorithm in content-centric networks. *Neurocomputing* **2015**, *149*, 1253–1269. [[CrossRef](#)]
13. Silla, C.N., Jr.; Koerich, A.L.; Kaestner, C.A. Feature selection in automatic music genre classification. In Proceedings of the 2008 Tenth IEEE International Symposium on Multimedia, Berkeley, CA, USA, 15–17 December 2008; pp. 39–44.
14. Cheng, G.; Ying, S.; Wang, B.; Li, Y. Efficient performance prediction for apache spark. *J. Parallel Distrib. Comput.* **2021**, *149*, 40–51. [[CrossRef](#)]
15. Karami, A. A framework for uncertainty-aware visual analytics in big data. In Proceedings of the 3rd International Workshop on Artificial Intelligence and Cognition (AIC) 2015, Turin, Italy, 28–29 September 2015; Volume 1510, pp. 146–155.
16. Karami, A.; Lundy, M.; Webb, F.; Boyajieff, H.R.; Zhu, M.; Lee, D. Automatic Categorization of LGBT User Profiles on Twitter with Machine Learning. *Electronics* **2021**, *10*, 1822. [[CrossRef](#)]
17. Meng, X.; Bradley, J.; Yavuz, B.; Sparks, E.; Venkataraman, S.; Liu, D.; Freeman, J.; Tsai, D.; Amde, M.; Owen, S.; et al. Mllib: Machine learning in apache spark. *J. Mach. Learn. Res.* **2016**, *17*, 1235–1241.
18. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]

19. Wibowo, F.W. Detection of Indonesian Dangdut Music Genre with Foreign Music Genres Through Features Classification Using Deep Learning. In Proceedings of the 2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), Jakarta, Indonesia, 29–30 January 2022; pp. 313–318.
20. Puppala, L.K.; Muvva, S.S.R.; Chinige, S.R.; Rajendran, P.S. A Novel Music Genre Classification Using Convolutional Neural Network. In Proceedings of the 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 8–10 July 2021; pp. 1246–1249.
21. Yuan, H.; Zheng, W.; Song, Y.; Zhao, Y. Parallel Deep Neural Networks for Musical Genre Classification: A Case Study. In Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 12–16 July 2021; pp. 1032–1035.
22. Feng, L.; Liu, S.; Yao, J. Music genre classification with paralleling recurrent convolutional neural network. *arXiv* **2017**, arXiv:1712.08370.
23. Kumar, K.; Chaturvedi, K. An Audio Classification Approach using Feature extraction neural network classification Approach. In Proceedings of the 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 28–29 February 2020; pp. 1–6.
24. Kobayashi, T.; Kubota, A.; Suzuki, Y. Audio feature extraction based on sub-band signal correlations for music genre classification. In Proceedings of the 2018 IEEE International Symposium on Multimedia (ISM), Taichung, Taiwan, 10–12 December 2018; pp. 180–181.
25. Chaudary, E.; Aziz, S.; Khan, M.U.; Gretschnann, P. Music Genre Classification using Support Vector Machine and Empirical Mode Decomposition. In Proceedings of the 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC), Karachi, Pakistan, 15–17 July 2021; pp. 1–5.
26. Pelchat, N.; Gelowitz, C.M. Neural network music genre classification. *Can. J. Electr. Comput. Eng.* **2019**, *43*, 170–173. [[CrossRef](#)]
27. Rong, F. Audio classification method based on machine learning. In Proceedings of the 2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Changsha, China, 17–18 December 2016; pp. 81–84.
28. Xavier, L.D.; Thirunavukarasu, R. A distributed tree-based ensemble learning approach for efficient structure prediction of protein. *Training* **2017**, *10*, 226–234. [[CrossRef](#)]
29. Köse, B.; Eken, S.; Sayar, A. Playlist generation via vector representation of songs. In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2016; pp. 179–185.
30. Eken, S. An exploratory teaching program in big data analysis for undergraduate students. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 4285–4304. [[CrossRef](#)]
31. Zeng, M.; Tan, X.; Wang, R.; Ju, Z.; Qin, T.; Liu, T.Y. Musicbert: Symbolic music understanding with large-scale pre-training. *arXiv* **2021**, arXiv:2106.05630.
32. Mehta, J.; Gandhi, D.; Thakur, G.; Kanani, P. Music Genre Classification using Transfer Learning on log-based MEL Spectrogram. In Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 8–10 April 2021; pp. 1101–1107.
33. Rahardwika, D.S.; Rachmawanto, E.H.; Sari, C.A.; Susanto, A.; Mulyono, I.U.W.; Astuti, E.Z.; Fahmi, A. Effect of Feature Selection on The Accuracy of Music Genre Classification using SVM Classifier. In Proceedings of the 2020 International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, Indonesia, 19–20 September 2020; pp. 7–11.
34. Kumar, D.P.; Sowmya, B.; Srinivasa, K. A comparative study of classifiers for music genre classification based on feature extractors. In Proceedings of the 2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Mangalore, India, 13–14 August 2016; pp. 190–194.
35. Khasgiwala, Y.; Tailor, J. Vision Transformer for Music Genre Classification using Mel-frequency Cepstrum Coefficient. In Proceedings of the 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), Kuala Lumpur, Malaysia, 24–26 September 2021; pp. 1–5.
36. Singh, Y.; Biswas, A. Robustness of musical features on deep learning models for music genre classification. *Exp. Syst. Appl.* **2022**, *199*, 116879. [[CrossRef](#)]
37. Folorunso, S.O.; Afolabi, S.A.; Owodeyi, A.B. Dissecting the genre of Nigerian music with machine learning models. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**. [[CrossRef](#)]
38. Xu, K.; Alif, M.A.; He, G. A novel music genre classification algorithm based on Continuous Wavelet Transform and Convolution Neural Network. In Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering, Xiamen, China, 22–24 October 2021; pp. 1269–1273.
39. Assefi, M.; Behraves, E.; Liu, G.; Tafti, A.P. Big data machine learning using apache spark MLlib. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 3492–3498.
40. Elbir, A.; Çam, H.B.; Iyican, M.E.; Öztürk, B.; Aydin, N. Music genre classification and recommendation by using machine learning techniques. In Proceedings of the 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), Adana, Turkey, 4–6 October 2018; pp. 1–5.
41. Sturm, B.L. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. *arXiv* **2013**, arXiv:1306.1461.
42. Cai, X.; Zhang, H. Music genre classification based on auditory image, spectral and acoustic features. *Multimed. Syst.* **2022**, 1–13. [[CrossRef](#)]

43. Li, T.; Ogihara, M.; Li, Q. A comparative study on content-based music genre classification. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, Toronto, ON, Canada, 28 July–1 August 2003; pp. 282–289.
44. Karunakaran, N.; Arya, A. A scalable hybrid classifier for music genre classification using machine learning concepts and spark. In Proceedings of the 2018 International Conference on Intelligent Autonomous Systems (ICoIAS), Singapore, 1–3 March 2018; pp. 128–135.
45. Elbir, A.; İlhan, H.O.; Serbes, G.; Aydın, N. Short Time Fourier Transform based music genre classification. In Proceedings of the 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT), Istanbul, Turkey, 18–19 April 2018; pp. 1–4.
46. Mayer, R.; Rauber, A. Musical genre classification by ensembles of audio and lyrics features. In Proceedings of the International Conference on Music Information Retrieval, Miami, FL, USA, 24–28 October 2011; pp. 675–680.
47. Devaki, P.; Sivanandan, A.; Kumar, R.S.; Peer, M.Z. Music Genre Classification and Isolation. In Proceedings of the 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 8–9 October 2021; pp. 1–6.