

## Article

# Attributes Relevance in Content-Based Music Recommendation System

Daniel Kostrzewa , Jonatan Chrobak and Robert Brzeski 

Department of Applied Informatics, Silesian University of Technology, 44-100 Gliwice, Poland;  
robert.brzeski@polsl.pl (R.B.)

\* Correspondence: daniel.kostrzewa@polsl.pl

**Abstract:** The possibility of recommendations of musical songs is becoming increasingly required because of the millions of users and songs included in online databases. Therefore, effective methods that automatically solve this issue need to be created. In this paper, the mentioned task is solved using three basic factors based on genre classification made by neural network, Mel-frequency cepstral coefficients (MFCCs), and the tempo of the song. The recommendation system is built using a probability function based on these three factors. The authors' contribution to the development of an automatic content-based recommendation system are methods built with the use of the mentioned three factors. Using different combinations of them, four strategies were created. All four strategies were evaluated based on the feedback score of 37 users, who created a total of 300 surveys. The proposed recommendation methods show a definite improvement in comparison with a random method. The obtained results indicate that the MFCC parameters have the greatest impact on the quality of recommendations.

**Keywords:** music recommendation system; music information retrieval; audio features; genre classification; song tempo; artificial neural network; MFCC; GTZAN



**Citation:** Kostrzewa, D.; Chrobak, J.; Brzeski, R. Attributes Relevance in Content-Based Music Recommendation System. *Appl. Sci.* **2024**, *14*, 855. <https://doi.org/10.3390/app14020855>

Academic Editors: Isabel Barbancho, Douglas O'Shaughnessy and Lorenzo J. Tardón

Received: 22 November 2023

Revised: 26 December 2023

Accepted: 17 January 2024

Published: 19 January 2024



**Copyright:** © 2024 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/>).

## 1. Introduction

Recommending systems [1–7] based on content and user preferences is a problem that frequently occurs in existing IT solutions. Any service in which data can be made public requires some way of sorting content. By customizing data served for users, higher retention can be achieved, and the application's usability is maximized. In the music domain, the complex nature of this topic is significant. Music streaming is increasingly common, and new services providing on-demand access to millions of songs are boosting daily. To attract new users, more and more advanced solutions in the field of music recommendation [8–10] are being implemented. However, the subjective nature of music tastes makes recommendations highly demanding. An audio signal contains many characteristics that affect the quality of received music. Even a slight change in one characteristic of a sound wave can affect the sound of an entire song.

Moreover, the human way of receiving sound must be considered during the analysis. Personal preferences in the context of music, which are natural to humans, in IT recommendation systems are immensely complex to define. However, a system that would provide new music precisely matched to personal preferences could be very advantageous to potential users. This problem requires careful analysis of many audio signal features and defining the similarity index between different songs. Therefore, investigating the gained recommendation mark based on individual song characteristics is an exciting and worthwhile issue from a scientific point of view.

The similarity in the context of music is very tricky to define. This difficulty originates from the high degree of subjectivity in human evaluation. Not all features of a piece of music can be accurately defined by numerical values, which could then be compared.

Admittedly, there are attributes for which creating a comparison is not a problem, such as a tempo or the key of a musical piece. However, in the case of characteristics such as genre or mood, the boundary between songs is not so obvious.

The musical genre [11] itself is an attractive feature in the context of song similarity. On the one hand, two songs from the same category will be more similar than those from totally different genres. On the other hand, finding an objective hierarchy of similarity between different musical genres is a non-trivial task. In addition, classifying a music piece [12,13] into a particular genre is often a very complex task, and the song itself may belong to several genres simultaneously. Since similarity can consist of many factors, it is challenging to find a metric or indicator to determine the similarity of one piece to another without conducting extensive research on a human test group. The problematic nature of this issue is well illustrated by the research conducted to date and existing solutions in the field of music recommendation.

In the research presented in this paper, the recommendation system is built using a probability function based on three factors: genre classification made by a neural network, Mel-frequency cepstral coefficients (MFCCs), and the tempo of the song. With the use of different combinations of the mentioned three features, four strategies were created. For a given strategy, using the calculated features and their weights, the distance to other songs is calculated. By putting in order these distances, the song that will be recommended is determined. The generated result of recommendation is evaluated based on the feedback score of 37 users, who created a total of 300 surveys.

The contribution of this paper is two-fold. Primarily, the content-based recommendation system is built with the use of different features extracted from music songs, namely Mel-frequency cepstral coefficients (MFCCs), the vector of the probability of the genre, and the tempo. Secondly, the relevance of different features in the scope of recommendation quality is measured by the mean opinion score.

The arrangement of the article is as follows. Sections 1 and 2 introduces the undertaken issue, provides an overview of recommendation systems and related works, and highlights the contribution. Section 3 describes in detail the built recommendation system. The conducted experiments are presented in Section 4. The paper is concluded in Section 5, where future work is also outlined.

## 2. Related Work

Recommendations of similar works can be divided into two main categories: knowledge-based recommendations [14] and content-based recommendations [15,16]. The knowledge-based approach focuses on external information available for a given work. This can be the song's metadata, including the author, a specific album, or genre and sub-genres. Statistics collected from system users are also often used, showing which songs were selected more often than others. By relying on external data, this strategy minimizes the subjectivity problem in song similarity. Using the data gathered from users, it does not have to define the similarity index but only matches its weights to the training data. What is more, using collaborative filtering, it is possible to generate a well-matched recommendation for a user at a low computational cost, using simple algorithms such as the K-nearest neighbors algorithm [15]. The generated recommendations feed themselves into the database, and the system becomes more accurate with use.

Unfortunately, using external data comes with some difficulties. Getting accurate data on users and their opinions is an expensive and time-consuming process, which can be a significant problem for new solutions that do not have a pre-existing user group. Recommendation results are entirely dependent on the amount and accuracy of the data, so if the information is limited or incomplete, the system will not work correctly. In addition, among the recommendations generated by the system, one can expect a preference to select more popular songs. By design, well-known songs among the user group will be more frequently selected by them, which will cause the model to assign them greater weight in further recommendations [15]. Another problem with this approach is the need for

expensive infrastructure. The requirement to store and process large amounts of external data may result in high memory cost in such a system. Taking advantage of collaborative filtering requires the creation of matrices matching users with recommendation objects. The matrices can be of considerable size for large user and song databases. In addition, as the system is used, its memory requirements will increase exponentially. Therefore, the knowledge-based recommendation strategy is most often used in large solutions with a pre-existing database of users and related statistics, for which large memory requirements are not a problem. For complete data, the results of solutions using this strategy may give better results than content-based solutions [17].

Unlike the knowledge-based strategy, the content-based strategy does not base the generated recommendations on external data but on the song's content. This approach does not need additional data because the characteristics are extracted from the object itself, which is a significant advantage. Based on the extracted parameters, a similarity classification is created. In the case of music songs, these parameters are obtained from the sound wave, e.g., using time-frequency analysis. Based on the collected data, comparing the deviations of individual parameters, it is possible to determine the similarity index between songs. This ensures that recommendations are not skewed in favor of more popular tracks, and the designed system is able to recommend new and niche results.

However, since the strategy bases its entire recommendation on a constructed similarity function [18], the problem of defining this concept is particularly highlighted here. It is difficult to account for and accurately record users' personal preferences. The selection of appropriate parameters and their weights is a crucial problem for the correct functioning of the system. In the case of music tracks, good results were obtained for Mel-frequency cepstral coefficients (MFCCs) [19]. The problem of matching weights and parameters can also be alleviated using the backpropagation of recommendation feedback. The song analysis and parameter extraction also have a higher computational requirement than the knowledge-based approach. Thus, this strategy is better suited for new solutions where data on users and their preferences are unavailable. With content analysis, it is possible to create recommendations without an extensive external database. In addition, even better results can be obtained using both strategies gaining advantages and minimizing the disadvantages of the aforementioned approaches [20].

Existing music recommendation systems often rely on multiple systems implementing both recommendation approaches, combining them into one working mechanism. One such solution is the Spotify platform with more than 80 million songs available, and more than 456 million users registered [21]. The primary goal of this service is to enable the streaming of music and podcasts. In addition to just being able to play music, the service has song recommendation features, such as *Discover Weekly*, which generates a playlist of new songs tailored to a user's preferences, or *Spotify Radio*, which generates a playlist based on a selected artist or group of songs.

The *Discover Weekly* feature uses collaborative filtering to generate new playlists. Based on the songs listened to by the user, the service prepares a preference profile and compares it to similar profiles of other users [22]. Based on this comparison, a list of songs repeated in playlists with similar profiles is prepared, which the user has not listened to yet.

The *Spotify Radio* feature uses a hybrid approach, combining both categories of recommendations. Using the EchoNest search engine, it retrieves both song content data (such as tempo or tonality) and external data (for example, popularity or user ratings) [23]. The found songs are fed back into the search engine system to generate more extended and diverse chains of similar songs. A user's positive or negative response automatically adjusts the system's score, propagating the playlist with new data similar to the favorite song or modifying the weights to improve recommendations in the future. By using the hybrid approach, recommendations can be more accurately matched to the song selected by the user.

However, many researchers are focused on extracting and comparing the features in finding similar songs [24,25]. The most common used ones are timbre, rhythm [26,27],

MFCCs [28,29], and spectrograms [19,30]. The extracted features can be compared in order to calculate the distance between songs. There are three main similarity measurements that are considered for this task [31]: K-means clustering, Expectation-Maximization with Monte Carlo Sampling, and Average Feature Vectors.

Recently, with the intense development of deep learning, many papers have been published on neural network-based music recommendation systems, including deep networks [32]. The main branches of explored architectures are: classic artificial neural networks [29,33,34], convolutional neural networks [19,35–39], different types of recurrent neural networks [33,34,40], and siamese neural networks [41].

### 3. Recommendation System

In this section, there is a detailed description of the built recommendation system along with the methods used.

#### 3.1. Considered Features

The metric for determining the similarity between songs in a built recommendation system is a function based on parameters extracted from the audio signal. The first group of parameters whose effectiveness has been investigated is the MFCCs of the song. These parameters approximate the human way of perceiving a piece of music by using the Mel scale and contain information about the timbre and texture of the musical piece. They were initially used in echo detection in seismic waves [42], but it has also found application in speech detection issues [43]. MFCCs have also been successfully applied to determine the genre and timbre of a musical piece [44]. Their usability has also been considered for digital signals [45]. In the field of music recommendation, their effectiveness in direct use has already been confirmed [19,46,47], so investigating their impact is essential in the context of the current wider study.

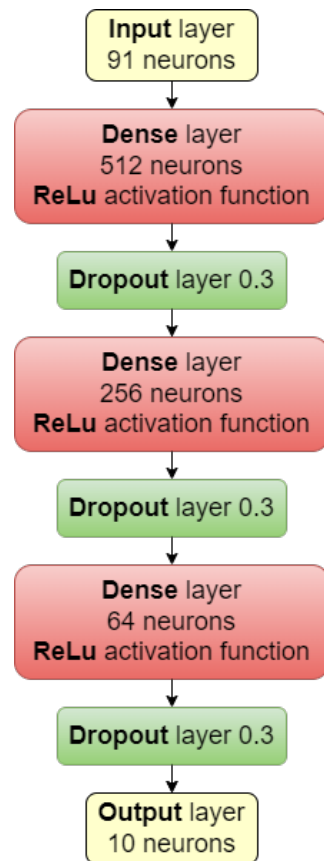
For the experiments, the values of 13 MFCCs were considered with a window size of 2048 frames, a sampling rate of 22,050 Hz, and a hop size of 512 samples. These values were calculated for three-second fragments of the song. Then, for ease of analysis and subsequent comparison of signals of different lengths and thus the number of segments, for each MFCC, the following statistical values were calculated: average value, minimum, maximum, median, standard deviation, skewness, and kurtosis. A total of 91 different parameters were obtained, creating a data vector of tested song for the MFCC feature. The implementation of the `librosa.feature.mfcc` function included in the Librosa [48] library for Python language was used to calculate the MFCC value for the segment. This function was used with default parameters.

Another feature whose influence on recommendation has been studied is the musical genre. This is a significant feature for the song similarity problem, but extracting it from the audio signal itself is not possible in a direct way, so this information is often contained in the song metadata, which the system does not use. However, genre classification can be obtained with reasonably good results, based on the values of MFCC parameters, using the artificial intelligence model [46,47]. For this reason, a multi-layer perceptron (MLP) model was used to extract the genre of the song.

The architecture of the developed neural network is shown in Figure 1. Data obtained from MFCC, i.e., the 91 mentioned values, are given to the network input. The network output is 10 neurons representing individual music genres, creating a data vector of tested song for the genre feature. The network was trained for 200 epochs. The classification quality for the used dataset after training the model is about 72% of the accuracy, which is a good result considering the difficulty of extracting this feature. Notably, the similarity analysis uses the entire vector of individual genre probabilities returned by the model, not just the most likely genre. Thanks to this, we have more data on the basis of which the results are developed.

The last feature used in the research is the tempo of the song. This creates the last single-element data vector of tested song for the tempo feature. Determining the tempo of

a song can also have a significant impact on similarity, as illustrated in previous research in this area [49]. All features (except the musical genre) are extracted using the Librosa library [48].



**Figure 1.** A multi-layer perceptron architecture used for genre classification.

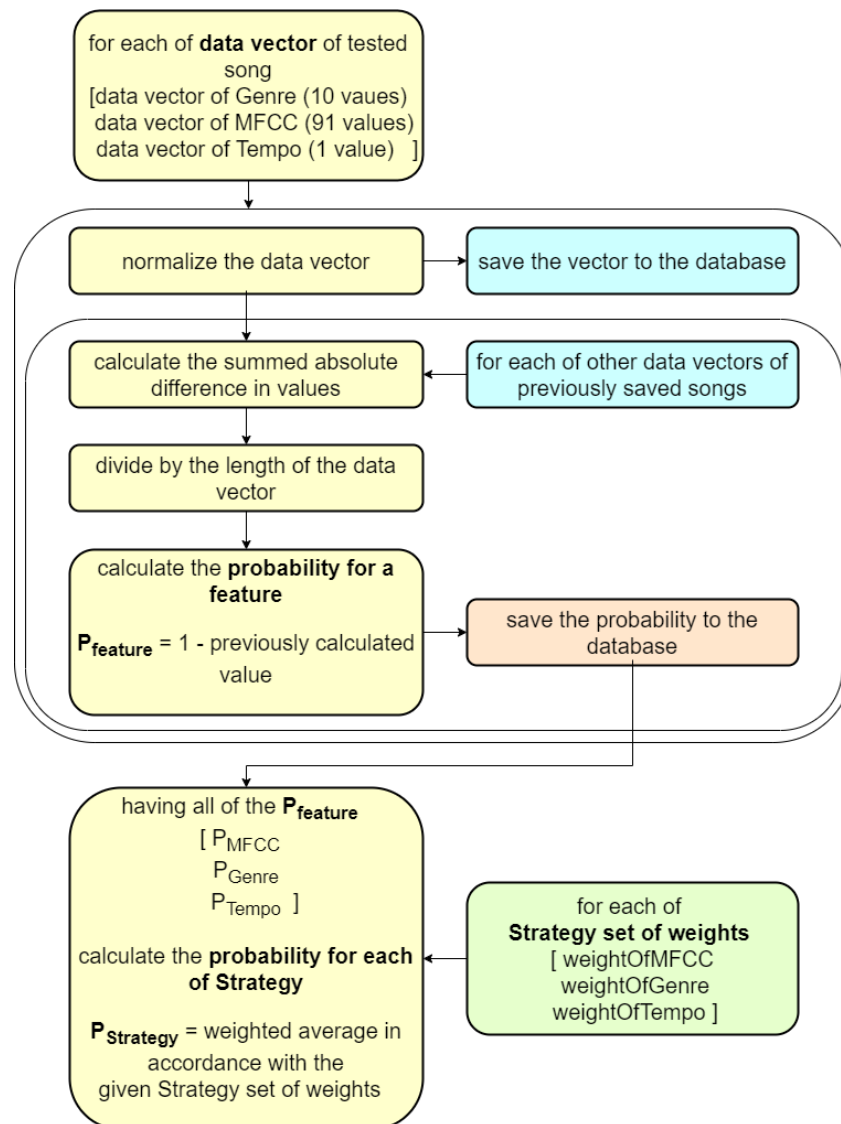
### 3.2. Recommendation Method

The system uses a probability function to generate recommendations (Figure 2). After selecting the input audio signal—the song that the user would like to add to the recommendation system, the system sorts the songs in the database, based on the likelihood ratio evaluation function maximizing its value. The result of the function is calculated based on the features obtained during the signal analysis. The values of the data vector for each of the features are normalized. In the case of features composed of multiple values (such as MFCCs or genre prediction vectors), each feature component is considered with the same weight. Then, for each of the other data vectors of previously saved songs, the summed absolute difference of all values between the tested data vector and the previously saved one is returned. The result is divided by the length of the data vector. Now, the probability of similarity of a song according to the given feature ( $P_{feature}$ ) can be calculated. It can be done by subtracting the obtained result from value one. In this way, a value from 0 to 1 is calculated for each feature, where 1 means that the compared musical pieces are the same. The calculated probabilities for the features ( $P_{MFCC}$ ,  $P_{Genre}$ ,  $P_{Tempo}$ ) are then used to calculate the final probability of similarity ( $P_{Strategy}$ ), according to the used strategy.

$$P_{strategy} = \frac{P_{MFCC} \cdot \text{weight of MFCC} + P_{Genre} \cdot \text{weight of Genre} + P_{Tempo} \cdot \text{weight of Tempo}}{\text{weight of MFCC} + \text{weight of Genre} + \text{weight of Tempo}}$$

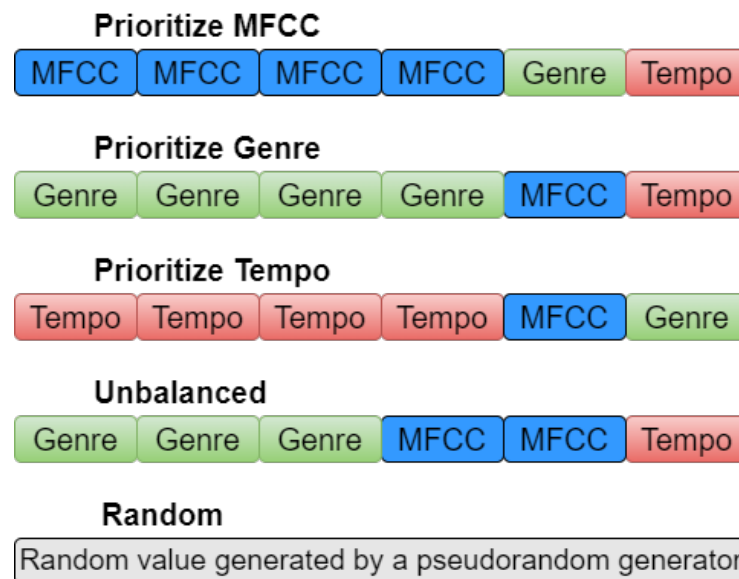
In this way, the main computational overhead is realized when adding a song to the recommendation system, In this moment the process of determining the values based on which the proximity of other musical songs is calculated. The final recommendation process—which is only searching for the closest song from those already added to the

database would make the final determination of recommended songs respectively fast and well scalable.



**Figure 2.** Calculating the probability of similarity of a song.

For the study, five sets of weights called strategies were prepared for the proposed features. Four strategies play the role of evaluating the relevance of individual features for the quality of recommendations. The strategies *PrioritizeMFCC*, *PrioritizeGenre* and *PrioritizeTempo* assign greater weight (4 times more weight) to one of the parameters (MFCCs, genre vector, and tempo respectively), giving equal consideration to the other features. The fourth strategy *Unbalanced* prioritizes genre vector first (3 times more weight), then MFCC values (2 times more weight), and finally tempo (single weight). Such a choice of individual weights was arbitrary and resulted from preliminary assumptions that the music genre may have the greatest impact on generating correct recommendations, and the tempo has the smallest impact. The fifth strategy does not consider any feature and returns a random value generated by a pseudorandom number generator. Its task is to evaluate the strategies created against the random solution, which is the study's control sample. The weights of each strategy were chosen arbitrarily based on the preliminary tests, and their distribution is shown in Figure 3. The number of individual blocks in a given strategy, indicates the strength/weight of a particular method in that strategy.



**Figure 3.** Distribution of weights for each strategy.

### 3.3. Dataset

The GTZAN dataset is one of the best-known and most widely used datasets in the field of music processing. It was created for music genre classification in 2002 and takes its name from the initials of its creator George Tzanetakis [44]. The set contains 1000 audio signals, which are 30 s excerpts of songs representing various musical genres. Each song is assigned to one of 10 musical genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. One hundred different songs are assigned to each genre. As a result, the data representation of each category is equal, making the dataset balanced.

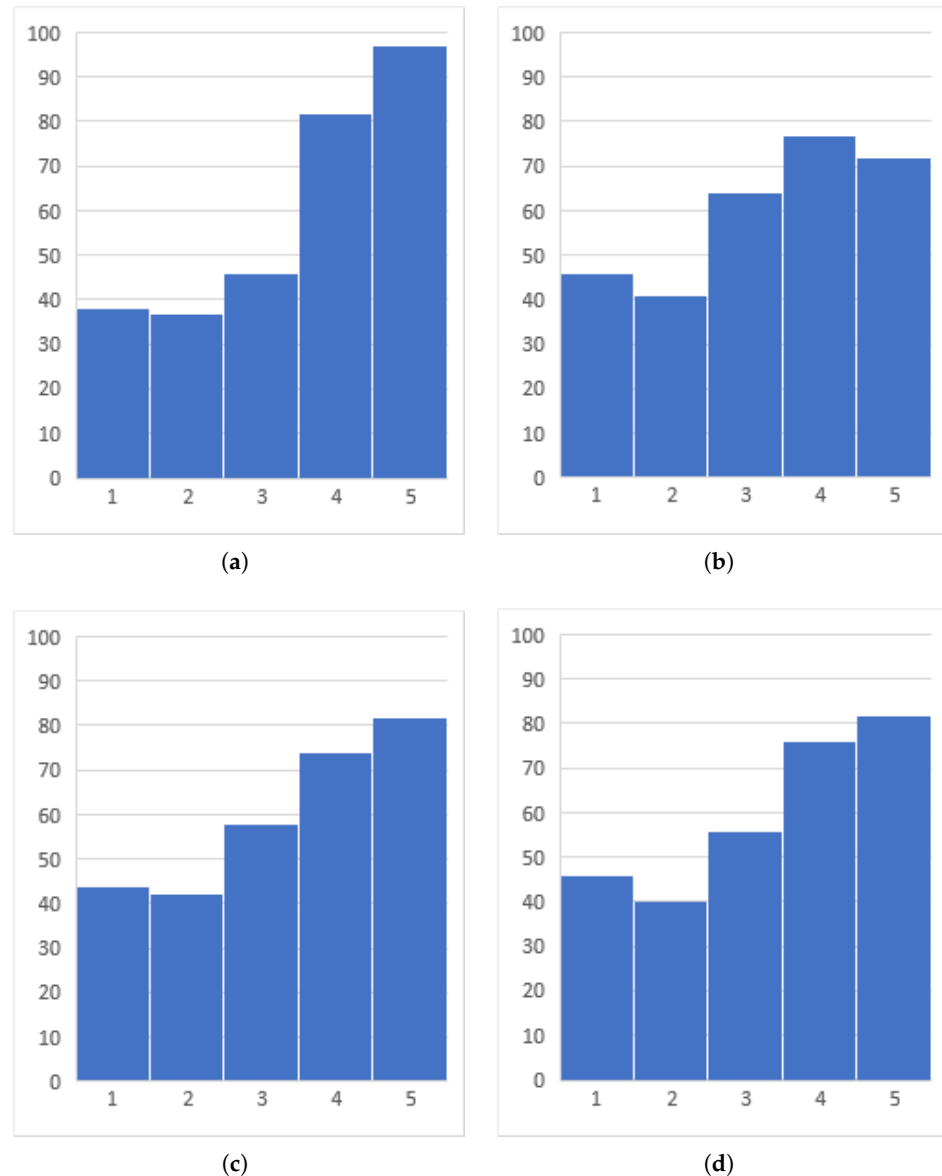
Larger datasets might offer a more thorough assessment of the recommendation system but at the same time, the disadvantages of these datasets will limit this evaluation. While it is easy to find a larger data set, finding a more appropriate one is not. The GTZAN collection contains 10 of the most typical, well-known music genres with a collection of files containing individual songs in musical form. This seems to be the most refined data set in terms of musical genres. Other collections either have fewer music genres, or these music genres do not necessarily represent the most popular ones, or the music pieces are not best suited to a given genre, or the database only contains metadata and the presented research requires music files.

### 3.4. Data Acquisition Method

The features used in the research have already been used in music recommendation tasks [19,20,47,49]. Thus, it can be assumed that their impact on the quality of recommendations will be significant, and the results will be much better than a random recommendation. Using a machine learning model to predict a song's music genre may sometimes generate erroneous outcomes, but this should not affect the recommendation results. Similar songs should generate comparable data, even if the original classification is inaccurate. In addition, using the full probability vector should allow better recommendations than if only the most likely genre criterion is used since similar songs should have similar probability vectors. Better results are also expected for musical genres with distinctive characteristics.

The complex nature of the recommendation makes it challenging to create an artificial quantifier of similarity since it is a largely subjective characteristic. Therefore, a purpose-built application with a recommendation evaluation system was used to collect data. During the study, 300 surveys (Figure 4) were collected from 37 different users. All of these users are adults of various age. The surveys were taken anonymously. During each survey, the user rated each recommendation on a scale of mark 1 to mark 5, assigning that the higher value of mark, the more similar the recommendation was to the given song. So,

mark is a subjective assessment of the similarity between songs given by the respondents. The indication as mark 5 is according to the user an ideal recommendation, and with mark 1 a completely unsuccessful recommendation. Recommendations were not rated against each other but only compared to the input one. The order in which recommendations were presented using different strategies was random. If two or more strategies pointed to the same song, it was presented only once in the survey.



**Figure 4.** Histograms of the distribution of recommendation ratings for selected strategies: (a) PrioritizeMFCC\_SimilarityStrategy, (b) PrioritizeGenre\_SimilarityStrategy, (c) PrioritizeTempo\_SimilarityStrategy, (d) Unbalanced\_SimilarityStrategy.

#### 4. Experimental Results

In this section, the description of conducted experiments are presented together with the presentation of the results obtained.

##### 4.1. Quantitative Results

Based on the collected surveys, an evaluation of the proposed recommendation strategies was carried out. For each strategy, 300 samples were collected. The control group

was generated by the *Random* strategy, which randomly generates recommendations. The collected data for each strategy was placed in Table 1.

**Table 1.** Summary of strategy recommendations and user evaluations.

| Strategy               | User Rating |             | Mark 1    |             | Mark 2    |             | Mark 3    |             | Mark 4    |             | Mark 5 |   | Avg.        |
|------------------------|-------------|-------------|-----------|-------------|-----------|-------------|-----------|-------------|-----------|-------------|--------|---|-------------|
|                        | No.         | %           | No.       | %           | No.       | %           | No.       | %           | No.       | %           | No.    | % |             |
| <i>Random</i>          | <b>215</b>  | <b>71.7</b> | <b>44</b> | <b>14.7</b> | 22        | 7.3         | 12        | 4.0         | 7         | 2.3         |        |   | 1.51        |
| <i>Unbalanced</i>      | 46          | 15.3        | 40        | 13.3        | 56        | 18.7        | 76        | 25.3        | 82        | 27.3        |        |   | 3.36        |
| <i>PrioritizeMFCC</i>  | 38          | 12.7        | 37        | 12.3        | 46        | 15.3        | <b>82</b> | <b>27.3</b> | <b>97</b> | <b>32.3</b> |        |   | <b>3.54</b> |
| <i>PrioritizeTempo</i> | 44          | 14.7        | 42        | 14.0        | 58        | 19.3        | 74        | 24.7        | 82        | 27.3        |        |   | 3.36        |
| <i>PrioritizeGenre</i> | 46          | 15.3        | 41        | 13.7        | <b>64</b> | <b>21.3</b> | 77        | 25.7        | 72        | 24.0        |        |   | 3.29        |

For each rating (value of mark), the result showing which strategy received the most such ratings (given by people checking the recommendation system) is in bold.

For the *Random* strategy, most recommendations received the marks of 1 or 2. For the 300 surveys collected, 86.33% of the user ratings were below the value of 3, with a significant predominance of the lowest value 1 of mark (215 surveys). Only seven times was the highest value of mark 5 selected with the use of this strategy. The mean value of the user rating (values of marks) for this strategy is 1.507, with a standard deviation of 0.959. This is the only strategy for which this trend occurred.

The remaining strategies had the most marks clustered in values between 3 and 5, and for none of them, the number of lowest ratings (values 1 of marks) exceeds the threshold of 16% of all ratings. The results among this group of strategies were very close to each other, as can be seen in Table 1. Remarkably similar are the strategies *PrioritizeGenre*, *PrioritizeTempo*, and *Unbalanced*. The distributions of values of marks for these strategies differ by less than 4%. The lowest mean value of mark of this group was obtained for the strategy of prioritizing a genre and was 3.293, with a standard deviation of 1.374. For the other two strategies mentioned, the mean value of mark was equal and was 3.36 with deviations of 1.394 for *PrioritizeTempo* and 1.404 for *Unbalanced*. The *PrioritizeMFCC* strategy deviates slightly from the norm. It has the highest number of marks of 4 and 5 of all strategies (82 and 97 surveys, respectively). This is also illustrated by the average value of mark of 3.543, with a standard deviation of 1.381.

The statistical significance tests were performed for the results obtained for the developed strategies. For each of the four non-random strategies, a statistical significance test was conducted with a  $\alpha = 0.05$  compared to the *Random* strategy. Similarly, analogous significance tests (Table 2) were conducted for the relationship between the *PrioritizeMFCC* strategy (which has the highest average value of marks from the surveys) and the other non-random strategies.

**Table 2.** Statistical significance tests for different strategies.

| Compared Strategies    |   |                        | <i>p</i> -Value       | Difference Statistically Significant |
|------------------------|---|------------------------|-----------------------|--------------------------------------|
| <i>PrioritizeMFCC</i>  | – | <i>Random</i>          | $4.8 \times 10^{-61}$ | Yes                                  |
| <i>PrioritizeTempo</i> | – | <i>Random</i>          | $1.8 \times 10^{-53}$ | Yes                                  |
| <i>PrioritizeGenre</i> | – | <i>Random</i>          | $1.5 \times 10^{-52}$ | Yes                                  |
| <i>Unbalanced</i>      | – | <i>Random</i>          | $1.7 \times 10^{-54}$ | Yes                                  |
| <i>PrioritizeMFCC</i>  | – | <i>PrioritizeTempo</i> | 0.0038                | Yes                                  |
| <i>PrioritizeMFCC</i>  | – | <i>PrioritizeGenre</i> | $5.9 \times 10^{-5}$  | Yes                                  |
| <i>PrioritizeMFCC</i>  | – | <i>Unbalanced</i>      | 0.00059               | Yes                                  |

Based on these results, it is evident that for each test, the *p* value is significantly below the  $\alpha$  value, proving that the obtained survey results are statistically significantly different.

#### 4.2. Impact of a Genre

The logical and most natural recommendation criterion seems to be the music genre. This is why the influence of the genres on the recommendations was additionally checked.

In this part of the research, the dependence of the strategy's average rating on the input song's genre was tested on the investigated data. Table 3 shows a summary of the developed strategies. The best average value of the mark for each music genre has been bolded. Most of the strategies behave similarly compared to the others, regardless of genre. There is a noticeable drop in performance for the country genre, with all strategies having an average value of a mark of about 2.3. Data analysis for this genre did not reveal an objective relationship between the indicators of individual parameters and lowered ratings. Therefore, this issue will be investigated in future research.

**Table 3.** Summary of music genres and average user ratings by strategy.

| Genres    | Strategy       |                 |                 |            | No. | Avg. |
|-----------|----------------|-----------------|-----------------|------------|-----|------|
|           | PrioritizeMFCC | PrioritizeTempo | PrioritizeGenre | Unbalanced |     |      |
| blues     | 3.97           | 3.52            | 3.45            | 3.58       | 31  | 3.63 |
| classical | 3.60           | 3.43            | 3.70            | 3.57       | 30  | 3.58 |
| country   | 2.35           | 2.23            | 2.31            | 2.31       | 26  | 2.30 |
| disco     | 3.63           | 3.41            | 3.38            | 3.25       | 32  | 3.41 |
| hiphop    | 3.59           | 3.78            | 3.22            | 3.33       | 27  | 3.48 |
| jazz      | 3.81           | 3.88            | 3.76            | 3.83       | 41  | 3.82 |
| metal     | 3.58           | 3.00            | 3.46            | 3.54       | 24  | 3.40 |
| pop       | 3.44           | 3.18            | 3.15            | 3.21       | 34  | 3.24 |
| reggae    | 3.87           | 3.80            | 3.07            | 3.47       | 30  | 3.55 |
| rock      | 3.32           | 2.96            | 3.16            | 3.24       | 25  | 3.17 |

The implementation of the recommendations for this genre, through the methods used, achieves clearly worse results, but at the current stage of the research, it cannot be clearly stated that this music genre is more difficult to recommend. Perhaps it is related to the taste of people carrying out the voting in the conducted research. In order to authenticate the difficulty of recommendations for a country music genre, tests with more people, including those who appreciate country music, should be conducted.

Table 3 also shows that the *PrioritizeMFCC* strategy achieved noticeably better results than the others when rating the blues genre, but the difference was insignificant.

The musical genre of the songs recommended by each strategy was also checked against the musical genre of the input song. A summary for each strategy can be seen in Table 4. It can be seen that for each of the strategies, there is a tendency to recommend songs of the same genre as the set song. Interestingly, the most genre-like recommendations were generated by the *PrioritizeMFCC* strategy; out of 300 results, 286 were recommended from the same category. The ratio is virtually identical for the other strategies, with 285 pieces from the same category for the *PrioritizeGenre* strategy, 283 for *Unbalanced*, and 277 for *PrioritizeTempo*.

**Table 4.** Comparison of the musical genre of the recommendation with the actual musical genre of the input song.

| Genres    | Strategy |           | PrioritizeTempo |           | PrioritizeGenre |           | Unbalanced |           |
|-----------|----------|-----------|-----------------|-----------|-----------------|-----------|------------|-----------|
|           | Same     | Different | Same            | Different | Same            | Different | Same       | Different |
| blues     | 31       | 0         | 31              | 2         | 31              | 1         | 31         | 1         |
| classical | 30       | 0         | 30              | 0         | 30              | 0         | 30         | 0         |
| country   | 25       | 2         | 23              | 5         | 23              | 1         | 23         | 1         |
| disco     | 30       | 5         | 27              | 3         | 30              | 3         | 30         | 6         |
| hiphop    | 22       | 0         | 24              | 2         | 22              | 0         | 22         | 0         |
| jazz      | 41       | 0         | 41              | 0         | 41              | 1         | 41         | 1         |
| metal     | 22       | 1         | 23              | 2         | 23              | 1         | 23         | 1         |
| pop       | 34       | 1         | 33              | 2         | 34              | 3         | 34         | 3         |
| reggae    | 27       | 1         | 26              | 3         | 26              | 3         | 26         | 2         |
| rock      | 24       | 4         | 19              | 4         | 25              | 2         | 23         | 2         |
| Sum       | 286      | 14        | 277             | 23        | 285             | 15        | 283        | 17        |

Analyzing Table 4, we can also see that the country genre does not suffer from any specific drop with respect to the other genres. It seems that the used methods and strategies work similarly for any music genre. This reinforces the idea that either country music is more diverse and therefore more difficult to recommend, or it depends on the taste of the people making the actual recommendation.

## 5. Conclusions and Future Work

The results obtained with the use of different features are much better rated by users than randomly returned musical songs. Based on the data, each strategy received an average mark above the value of 3 during the evaluation, while the average random recommendation received an average value of mark of 1.51. Even with a not so big research dataset, this trend is very clearly observed. In addition, the average user rating of 3.54 (average value of mark) for the best strategy is promising, especially for a new system using only a content-based approach. The recommendation marks did not differ significantly across music genres, except for the country music genre. However, the small amount of data and additional analysis of the results indicates that this may have been an outlier observation. If more data had been collected, possibly, the value would have come closer to the others. This idea is also supported by the fact that this trend occurred for each strategy, despite the significant differences in the included weights for each feature. Enlarging the set of collected data and using it again to adjust the weights of the strategies could yield even better results.

Since a relatively small amount of data was collected in this research, it is challenging to draw decisive conclusions. The trends expected in the research are visible, and the features extracted during the analysis significantly impact the quality of the recommendations. The data obtained indicate that the MFCC parameters have the greatest impact on the quality of recommendations. The *PrioritizeMFCC* strategy, which gives the most significant weight to this feature, achieved the best results among the surveyed users. This difference is evident with the data obtained and occurs consistently, regardless of other factors such as genres. However, further research is required to determine how much impact this indicator has on the quality of recommendations.

The solution has great potential for development, and its extension would provide an accurate basis for use in music recommendation. Potential development paths for the recommendation system include expanding audio feature analysis, enlarging the number of features, and increasing the number of available strategies. Better results could be achieved by using the research data to adjust the weights assigned to the parameters, for example, by creating strategies based on an artificial intelligence model with backward error propagation.

The research obtained with the proposed recommendation strategies allows preliminary conclusions to be drawn, and an increase in the number of surveys is needed to develop the research. The recommendation system will need to be expanded to obtain more information. Testing the system with other data sets, such as the Free Music Archive dataset, would also be beneficial. Another research direction would be to implement a larger number of more diverse strategies. This would allow new parameters to be explored, non-linear relationships between them to be investigated, and similarity scores to be tested. In addition, creating strategies based on, for example, deep neural networks, would make it possible to adjust the weights of individual features as the recommendation system runs, based on user feedback. Nevertheless, the presented results are very promising and encourage further work.

**Author Contributions:** Conceptualization, D.K. and J.C.; methodology, D.K.; software, J.C.; validation, D.K., J.C. and R.B.; data curation, J.C.; writing—original draft preparation, D.K. and R.B.; writing—review and editing, D.K. and R.B.; supervision, D.K.; funding acquisition, D.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by the Statutory Research Fund of Department of Applied Informatics, Silesian University of Technology, Gliwice, Poland.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset used in this research is freely available on the GTZAN Dataset <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification> (accessed on 20 November 2023).

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

## References

1. Ko, H.; Lee, S.; Park, Y.; Choi, A. A survey of recommendation systems: Recommendation models, techniques, and application fields. *Electronics* **2022**, *11*, 141. [CrossRef]
2. Isinkaye, F.O.; Folajimi, Y.O.; Ojokoh, B.A. Recommendation systems: Principles, methods and evaluation. *Egypt. Inform. J.* **2015**, *16*, 261–273. [CrossRef]
3. Naumov, M.; Mudigere, D.; Shi, H.J.M.; Huang, J.; Sundaraman, N.; Park, J.; Wang, X.; Gupta, U.; Wu, C.J.; Azzolini, A.G.; et al. Deep learning recommendation model for personalization and recommendation systems. *arXiv* **2019**, arXiv:1906.00091.
4. Ebesu, T.; Shen, B.; Fang, Y. Collaborative memory network for recommendation systems. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 515–524.
5. Singhal, A.; Sinha, P.; Pant, R. Use of deep learning in modern recommendation system: A summary of recent works. *arXiv* **2017**, arXiv:1712.07525.
6. Rumiantcev, M.; Khriyenko, O. Emotion based music recommendation system. In Proceedings of the Conference of Open Innovations Association FRUCT, Trento, Italy, 7–9 September 2020.
7. Chavare, S.R.; Awati, C.J.; Shirgave, S.K. Smart recommender system using deep learning. In Proceedings of the 2021 IEEE 6th international conference on inventive computation technologies (ICICT), Coimbatore, India, 20–22 January 2021; pp. 590–594.
8. Fessahaye, F.; Perez, L.; Zhan, T.; Zhang, R.; Fossier, C.; Markarian, R.; Chiu, C.; Zhan, J.; Gewali, L.; Oh, P. T-recsys: A novel music recommendation system using deep learning. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 11–13 January 2019; pp. 1–6.
9. Mahadik, A.; Milgir, S.; Patel, J.; Jagan, V.B.; Kavathekar, V. Mood based music recommendation system. *Int. J. Eng. Res. Technol. (IJERT)* **2021**, *10*, 553–559.
10. Wang, R.; Ma, X.; Jiang, C.; Ye, Y.; Zhang, Y. Heterogeneous information network-based music recommendation system in mobile networks. *Comput. Commun.* **2020**, *150*, 429–437. [CrossRef]
11. Aucouturier, J.J.; Pachet, F. Representing musical genre: A state of the art. *J. New Music Res.* **2003**, *32*, 83–93. [CrossRef]
12. Heakl, A.; Abdelgawad, A.; Parque, V. A Study on Broadcast Networks for Music Genre Classification. In Proceedings of the 2022 IEEE International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8.
13. Xu, Z.; Feng, Y.; Song, S.; Xu, Y.; Wang, R.; Zhang, L.; Liu, J. Research on Music Genre Classification Based on Residual Network. In Proceedings of the International Conference on Mobile Computing, Applications, and Services, Messina, Italy, 17–18 November 2022; pp. 209–223.
14. Oramas, S.; Nieto, O.; Sordo, M.; Serra, X. A deep multimodal approach for cold-start music recommendation. In Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems, Como, Italy, 27 August 2017; pp. 32–37.
15. Madathil, M. *Music Recommendation System Spotify-Collaborative Filtering*; Reports in Computer Music; Aachen University: Aachen, Germany, 2017.
16. Niyazov, A.; Mikhailova, E.; Egorova, O. Content-based music recommendation system. In Proceedings of the 2021 IEEE 29th Conference of Open Innovations Association (FRUCT), Madrid, Spain, 13–15 June 2011; pp. 274–279.
17. Slaney, M. Web-Scale Multimedia Analysis: Does Content Matter? *IEEE Multimed.* **2011**, *18*, 12–15. [CrossRef]
18. Ontañón, S. An overview of distance and similarity functions for structured data. *Artif. Intell. Rev.* **2020**, *53*, 5309–5351. [CrossRef]
19. Van den Oord, A.; Dieleman, S.; Schrauwen, B. Deep content-based music recommendation. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 1–9.
20. McFee, B.; Barrington, L.; Lanckriet, G. Learning Content Similarity for Music Recommendation. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 2207–2218. [CrossRef]
21. About Spotify. Available online: <https://newsroom.spotify.com/company-info/> (accessed on 20 November 2023).
22. The Magic That Makes Spotify’s Discover Weekly Playlists so Damn Good. Available online: <https://qz.com/571007/the-magic-that-makes-spotifys-discover-weekly-playlists-so-damn-good> (accessed on 20 November 2023).
23. Germain, A.; Chakareski, J. Spotify Me: Facebook-assisted automatic playlist generation. In Proceedings of the 2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP), Pula, Italy, 30 September–2 October 2013; pp. 25–28. [CrossRef]
24. Bogdanov, D.; Serrà, J.; Wack, N.; Herrera, P.; Serra, X. Unifying low-level and high-level music similarity measures. *IEEE Trans. Multimed.* **2011**, *13*, 687–701. [CrossRef]

25. Mak, C.M.; Lee, T.; Senapati, S.; Yeung, Y.T.; Lam, W.K. Similarity Measures for Chinese Pop Music Based on Low-level Audio Signal Attributes. In Proceedings of the International Society for Music Information Retrieval Conference (ISMIR 2010), Utrecht, The Netherlands, 9–13 August 2010; pp. 513–518.
26. Bogdanov, D.; Boyer, H. How much metadata do we need in music recommendation? A subjective evaluation using preference sets. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), Miami, FL, USA, 24–28 October 2011; Klapuri, A., Leider, C., Eds.; International Society for Music Information Retrieval (ISMIR): Montréal, QC, Canada, 2011; pp. 97–102.
27. Cano, P.; Koppenberger, M.; Wack, N. An industrial-strength content-based music recommendation system. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, 15–19 August 2005; p. 673.
28. Hoffmann, P.; Kaczmarek, A.; Spaleniak, P.; Kostek, B. Music recommendation system. *J. Telecommun. Inf. Technol.* **2014**, 59–69.
29. Liang, D.; Zhan, M.; Ellis, D.P. Content-Aware Collaborative Music Recommendation Using Pre-trained Neural Networks. In Proceedings of the International Society for Music Information Retrieval Conference, Malaga, Spain, 26–30 October 2015; pp. 295–301.
30. Wang, X.; Wang, Y. Improving content-based and hybrid music recommendation using deep learning. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 627–636.
31. Magno, T.; Sable, C. A Comparison of Signal Based Music Recommendation to Genre Labels, Collaborative Filtering, Musicological Analysis, Human Recommendation and Random Baseline. In Proceedings of the International Society for Music Information Retrieval Conference, Philadelphia, PA, USA, 14–18 September 2008; pp. 161–166.
32. Schedl, M. Deep learning in music recommendation systems. *Front. Appl. Math. Stat.* **2019**, *5*, 44. [\[CrossRef\]](#)
33. Sachdeva, N.; Gupta, K.; Pudi, V. Attentive neural architecture incorporating song features for music recommendation. In Proceedings of the 12th ACM Conference on Recommender Systems, Vancouver, BC, Canada, 2 October 2018; pp. 417–421.
34. Lin, Q.; Niu, Y.; Zhu, Y.; Lu, H.; Mushonga, K.Z.; Niu, Z. Heterogeneous knowledge-based attentive neural networks for short-term music recommendations. *IEEE Access* **2018**, *6*, 58990–59000. [\[CrossRef\]](#)
35. Vall, A.; Dorfer, M.; Eghbal-Zadeh, H.; Schedl, M.; Burjorjee, K.; Widmer, G. Feature-combination hybrid recommender systems for automated music playlist continuation. *User Model. User-Adapt. Interact.* **2019**, *29*, 527–572. [\[CrossRef\]](#)
36. Abdul, A.; Chen, J.; Liao, H.Y.; Chang, S.H. An emotion-aware personalized music recommendation system using a convolutional neural networks approach. *Appl. Sci.* **2018**, *8*, 1103. [\[CrossRef\]](#)
37. Zhang, Y. Music recommendation system and recommendation model based on convolutional neural network. *Mob. Inf. Syst.* **2022**, *2022*, 3387598. [\[CrossRef\]](#)
38. Dong, Y.; Guo, X.; Gu, Y. Music recommendation system based on fusion deep learning models. *J. Phys. Conf. Ser.* **2020**, *1544*, 012029. [\[CrossRef\]](#)
39. Elbir, A.; Aydin, N. Music genre classification and music recommendation by using deep learning. *Electron. Lett.* **2020**, *56*, 627–629. [\[CrossRef\]](#)
40. Vall, A.; Quadrana, M.; Schedl, M.; Widmer, G. The importance of song context and song order in automated music playlist generation. *arXiv* **2018**, arXiv:1807.04690.
41. Pulis, M.; Bajada, J. Siamese Neural Networks for Content-based Cold-Start Music Recommendation. In Proceedings of the 15th ACM Conference on Recommender Systems, Amsterdam, The Netherlands, 27 September–1 October 2021; pp. 719–723.
42. Bogert, B.P.; Ossanna, J. Computer Experimentation on Echo Detection, Using the Cepstrum and Pseudoautocovariance. *J. Acoust. Soc. Am.* **1966**, *39*, 1258–1259. [\[CrossRef\]](#)
43. Stern, R.M.; Acero, A. *Acoustical Pre-Processing for Robust Speech Recognition*; Technical Report; Carnegie-Mellon University Pittsburgh PA School of Computer Science: Pittsburgh, PA, USA, 1989.
44. Tzanetakis, G.; Cook, P. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.* **2002**, *10*, 293–302. [\[CrossRef\]](#)
45. Sigurdsson, S.; Petersen, K.B.; Lehn-Schiøler, T. Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music. In Proceedings of the International Conference on Music Information Retrieval, Victoria, BC, Canada, 8–12 October 2006; pp. 286–289.
46. Kostrzewa, D.; Ciszynski, M.; Brzeski, R. Evolvable hybrid ensembles for musical genre classification. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Boston, MA, USA, 9–13 July 2022; pp. 252–255.
47. Kostrzewa, D.; Mazur, W.; Brzeski, R. Wide Ensembles of Neural Networks in Music Genre Classification. In Proceedings of the Computational Science–ICCS 2022: 22nd International Conference, London, UK, 21–23 June 2022; Proceedings, Part II; Springer: Berlin/Heidelberg, Germany, 2022; pp. 64–71.
48. Librosa Library. Available online: <https://librosa.org> (accessed on 20 November 2023).
49. Zhu, X.; Shi, Y.Y.; Kim, H.G.; Eom, K.W. An integrated music recommendation system. *IEEE Trans. Consum. Electron.* **2006**, *52*, 917–925. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.