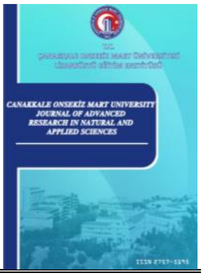
Çanakkale Onsekiz Mart University

Journal of Advanced Research in Natural and Applied Sciences

dergipark.org.tr/tr/pub/jarnas

Open Access



e-ISSN: 2757-5195 2024, Vol. 10, Issue 2, Pages: 312-328 doi.org/10.28979/jarnas.1392427

Machine Learning-Based Classification of Turkish Music for Mood-Driven Selection

Nazime Tokgöz[1], Ali Değirmenci[2], Ömer Karal[3]

[1,2,3]Department of Electrical and Electronics Engineering, Faculty of Engineering, Ankara Yıldırım Beyazıt University, Ankara, Türkiye

Article History

Received: 06 Oct 2023

Accepted: 01 Feb 2024

Published: 25 Jun 2024

Research Article

Abstract − Music holds a significant role in our daily lives, and its impact on emotions has been a focal point of research across various disciplines, including psychology, sociology, and statistics. Ongoing studies continue to explore this intriguing relationship. With advancing technology, the ability to choose from a diverse range of music has expanded. Recent trends highlight a growing preference for searching for music based on emotional attributes rather than individual preferences or genres. The act of selecting music based on emotional states is important on both a universal and cultural level. This study seeks to employ machine learning-based methods to classify four different music genres using a minimal set of features. The objective is to facilitate the process of choosing Turkish music according to one's mood. The classification methods employed include Decision Tree, Random Forest (RF), Support Vector Machines (SVM), and k-Nearest Neighbor, coupled with the Mutual Information (MI) feature selection algorithm. Experimental results reveal that, with all features considered in the dataset, RF achieved the highest accuracy at 0.8098. However, when the MI algorithm was applied, SVM exhibited the best accuracy at 0.8068. Considering both memory consumption and accuracy, the RF method emerges as a favorable choice for selecting Turkish music based on emotional states. This research not only advances our understanding of the interaction between music and emotions but also provides practical insights for individuals who want to shape their music according to their emotional preferences.

Keywords − *Classification, emotions, feature selection, music genres, mutual information*

1. Introduction

Music, an integral part of our daily lives, transcends cultural and linguistic boundaries, serving various purposes with its universal language and structure. From melodies heard in the womb, comforting lullabies sung as a baby, to songs shared during school days, music plays an important role in shaping our experiences and emotions [1, 2].

Often referred to as the "food of the soul," music possesses the transformative power to influence a person's emotional state in response to external stimuli. Recent years have seen music increasingly used in meditation practices and subliminal suggestions, emphasizing the potential impact of music on psychological health. [3]. The suggestive power of music holds particular significance in therapeutic contexts, where repetition enhances the acceptance of suggestions, with music serving as a carrier of hidden messages. The choice of songs and lyrics can wield a profound influence on emotions, whether intentional or not, eliciting calming, instructive, exhilarating, or even angering effects based on musical type and rhythm. People usually make predictions about what will happen next while listening to music. This causes music to be processed in the brain as action, emotion, and learning [4]. The emotional and calming effects of music have been evidenced in diverse settings,

[1]n.tokgoz@ankarad.com; [2]alidegirmenci@ybu.edu.tr; [3]omerkaral@ybu.edu.tr (Corresponding Author)

from studies involving pregnant women to those conducted before and during surgical procedures [5].

With music's profound impact on individual moods and its listening by large audiences, there has been a growing inclination for people to choose music aligned with their emotional states. This desire has prompted the need to develop methods for selecting or measuring the change in emotion based on music selection [6].

In recent years, numerous studies have been conducted around the world exploring the intersection of music and emotions. Fritz et al. recorded individuals' facial expressions while listening to music and then segmented their moods into three basic emotional states: happy, sad, and scared [7]. Mahadik et al. focused on classifying music selections based on facial emotion expressions, including emotions such as happy, angry, sad, neutral, surprised, and afraid, and then recommending music that aligned with the identified emotion [8]. Durahim et al. developed a model to automatically detect perceived emotion from song lyrics with the help of machine learning algorithms [9]. Er and Aydilek pioneered a novel music emotion recognition method, employing deep learning techniques on pre-trained deep networks, diverging from frequently utilized machine learning methods [10]. The optimal result, determined through VGG-16 in the Fc7 layer, yielded an accuracy of 89.2%. Chaudhary et al. proposed three distinct Music Emotion Classification Systems (MECS), with the first two utilizing Convolutional Neural Network (CNN) and the last employing Support Vector Machine (SVM) [11]. The song database was categorized into four, eight, and sixteen classes, achieving accuracies of 91%, 88%, and 86% in three experiments for the first MECS model. Quasim et al. introduced an Emotion-Based Music Recommendation and Classification Framework (EMRCF), achieving 96.12% accuracy with high precision in classifying songs based on individuals' interpersonal teams, incorporating memory and emotion [12]. Su et al. proposed a music recognition method combining Deep Learning (DL) and SVM, demonstrating its superiority over other audio-based music emotion tagging methods based on results obtained from the CAL500 dataset [13]. Pandrea et al. conducted experimental studies on a novel emotion detection approach, presenting a language-aware end-to-end architecture (SincNet) model that learns to label emotions in music with lyrics in three different languages, achieving 71% accuracy on the Turkish Emotion dataset [14]. Ciborowski et al. introduced an emotion model predicting nine emotional states, with color assignments based on the color theory in film, achieving the best result with the Inception V3 method with a minimal Mean Squared Error (MSE) of 0.0542% [15]. Huang et al. developed an end-to-end Attention-based Deep Feature Fusion (ADFF) technique for music emotion recognition, achieving relative improvements of 10.43% and 4.82% in valence [16]. Zhang et al. employed VGGish and SVM methods together for emotion classification, obtaining the highest success with 66.98% accuracy [17].

In the realm of machine learning, feature selection is crucial for revealing the most descriptive features (k features) while minimizing generalization error. This process significantly influences the performance of machine learning models, as an excess of features can extend training times, reduce interpretability, and potentially lead to overfitting. Due to overfitting, the model success may be high in the training but low in the test data.

The Turkish Music Emotion data set in Er and Aydilek used in our study has been the focus of some studies in the literature [10]. Moldovan developed the Binary Horse Optimization Algorithm (BHOA), a bio-inspired feature selection method that can be used for different classification techniques. The performance of BHOA was evaluated via six machine learning methods (Logistic Regression (LR), k-Nearest Neighbors (kNN), Decision Tree (DT), Random Forest (RF), Gradient Boosted Tree (GBT), and SVM) on nine different datasets, including the Turkish Music Emotion dataset [18]. The datasets were randomly split into 80% (training) and 20% (testing) sets, and the results from BHOA were compared with the binary particle swarm optimization, binary gray wolf optimizer, and binary crow search optimization algorithms. Experimental results showed that the developed BHOA showed promising performance when compared with alternative optimization algorithms. Pandrea et al. applied the SincNet architecture to Mandarin, English, and Turkish music emotion datasets with in-dataset and mixed dataset setups [14]. The accuracy achieved within the scope of Turkish Music Emotion data was reported as 63%. Feng et al. proposed the FESVM method, which is SVM and feature generation-based ensemble learning [19]. New features are produced from the classification probabilities obtained from the basic classifiers, and hyperparameter optimization is carried out with 5-fold cross-validation, using 70% of the datasets for training and 30% for testing. The accuracy of the FESVM method on the Turkish Emotion dataset is reported as 81.5%. Compared with the results obtained in our study, the combination of our proposed MI feature selection method and RF machine learning algorithm achieved an accuracy of 80.75% using only 20 features. This shows that the method we present is more convenient in terms of both memory and speed.

In this research, the Turkish Music dataset from Er and Aydilek served as the foundation for selecting music based on individual moods or gauging the emotional shifts prompted by specific music choices [10]. To achieve this objective, four distinct machine-learning algorithms were employed: DT, kNN, RF, and SVM. Diverging from conventional practices, the dataset was not partitioned into fixed proportions for training and testing; instead, the k-fold cross-validation method was adopted to enhance reliability and observe the impact of each sample. To optimize the performance of kNN, RF, and SVM methods, meticulous investigations into method-specific hyperparameters were conducted. The grid search method was utilized to identify hyperparameters that yielded the highest scores. Subsequently, the importance of each feature was assessed through the Mutual Information (MI) feature selection algorithm. Machine learning models were trained using subsets of data, with features ranked from the highest to the lowest mutual information score. This approach aimed to achieve optimal performance by utilizing the minimum number of features. Experimental results indicated that the RF method outperformed others in both feature selection and without feature selection. The RF method achieved the highest accuracy score of 81% when considering all features in the dataset. Impressively, even with only 20 features, the accuracy remained high at 80.75%. Notably, thanks to the MIFS algorithm, a comparable level of accuracy was attained using only 40% of the features in the dataset. This highlights the effectiveness of feature selection techniques in optimizing the performance of machine learning models.

The subsequent sections of this study are outlined as follows: In Section 2, detailed presentation of the materials used, and the methodologies employed is provided. Section 3 delves into the outcomes of the study and engages in a comprehensive discussion. The MI feature selection algorithm, instrumental in identifying descriptive features within the dataset, is introduced. Additionally, the machine learning methods employed for music emotion classification are explicated, paving the way for an in-depth exploration of the results. Section 4 encompasses a thorough examination of the metrics employed to evaluate the performance of machine learning methods in Turkish music emotion classification. This includes an analysis of both machine learning methods and the feature selection technique, shedding light on the experimental results. The final section encapsulates the conclusions drawn from the study and outlines potential avenues for future research. It provides a succinct summary of the key findings and offers insights into areas where further exploration and refinement could enhance the understanding and application of machine learning in the context of Turkish music emotion classification.

2. Materials and Methods

In this study, Turkish music emotion datasets were used from the UCI machine learning repository. The dataset, structured as a discrete model, comprises four distinct classes representing basic emotional states, namely happy, sad, angry, and relaxed. To create the dataset, verbal, and non-verbal music from diverse genres of Turkish music were carefully selected. The database includes a total of 100 music pieces, ensuring an equal distribution of samples across each emotional class. The original dataset has 400 samples, each lasting 30 seconds [10]. For feature extraction, the MIR toolbox was utilized to analyze the emotional content within musical signals. Various features, including mel frequency cepstral coefficients, chromagram, tempo, and spectral and harmonic features, were extracted. This comprehensive feature extraction process aimed to provide a nuanced examination of the emotional nuances embedded in the Turkish music samples.

2.1. Methods

Within the framework of this study, a comparative analysis is conducted on four distinct machine learning methods employed to classify emotions within the Turkish music emotion dataset. Additionally, feature selection is integrated into the methodology to pinpoint the most informative features, as explained by Çakır et al. [20]. The following subsections provide details regarding the specific machine learning methods and feature selection techniques, ensuring a comprehensive understanding of the methodology employed in the study. This rigorous approach aims to enhance the interpretability and effectiveness of the machine learning models developed for emotion classification in Turkish music.

2.1.1. Mutual Information (MI)

The dependency or measure of shared information between two random variables is defined as MI. The definition of entropy can be defined by the concept given by Shannon [21].

$$H(A) := -\sum_{x} p(a) \log_2\big(p(a)\big) \tag{2.1}$$

(2.1) gives the uncertainty associated with the random variable A. In feature selection, the focus is on maximizing the information shared between the target and feature variables.

The joint entropy value, defined by (2.2), quantifies the uncertainty present in two random discrete variables, like A and B, simultaneously.

$$\begin{aligned} H(B\backslash A) &= -\sum_{a\in A}\sum_{b\in B} p(a,b) \log_2 p(b\backslash a) \\ H(A,B) &= -\sum_{a\in A}\sum_{b\in B} p(a,b) \log_2 p(a,b) \end{aligned} \tag{2.2}$$

When the specific value of A is denoted as "a" and the specific value of B is "b", the probability of these values occurring together, denoted as $p(a,b)$, is taken into account. The expression $p(a,b) \log_2 p(a,b)$ is defined to be 0 if $p(a,b) = 0$. The value of joint entropy varies based on the dependence between A and B. If A and B are completely dependent, then the joint entropy is at its minimum, whereas it reaches its maximum when they are entirely independent.

Conditional entropy measures the uncertainty of B when the value of A is known and is mathematically defined as follows (2.3):

$$H(B\backslash A) = -\sum_{a\in A}\sum_{b\in B} p(a,b)\log_2 p(b\backslash a) \tag{2.3}$$

Entropy serves as a metric to quantify the information that one variable contains about another. MI is defined as the relative entropy between joint distributions, as outlined by Gonzalez-Lopez et al. [22]. The product distribution is depicted in (2.4):

$$MI(A;B) = \sum_{a\in A}\sum_{b\in B} p(a,b)\log\left(\frac{p(a,b)}{p(a),p(b)}\right) \tag{2.4}$$

$$MI(A,B) = H(B) - H(B|A)$$

$$MI(A,B) = H(A) - H(A|B) \tag{2.5}$$

$$MI(A,B) = H(A) + H(B) - H(A,B)$$

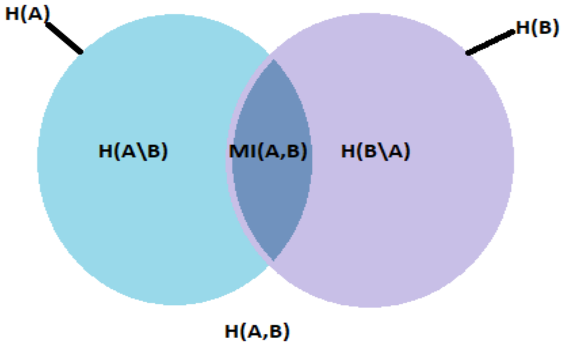


Figure 1. Venn diagram illustrating the entropy and MI relationships between two correlated variables (A and B)

An algorithm is used to determine maximum of MI. There is a subset of features initialized with a feature, denoted by the S matrix, and features are sequentially added to this subset one by one Zhang et al. [23].

$$j = \underset{j\notin S}{\operatorname{argmax}}\, MI\left(Y_S \cup x^j; t\right) \tag{2.6}$$

(2.6) is also employed for the selection of the initial feature. The chosen features are treated as independent. Once the increase in MI reaches its peak, the addition of new features is stopped. When the increase in MI is highest, adding new features is stopped. This methodology facilitates the reduction of feature dimensionality [24].

2.1.2. k-Nearest Neighbour (kNN)

The kNN is a straightforward yet widely used machine learning technique that has demonstrated success across various fields. Its application extends to diverse data types such as free text, images, audio, and video. In the kNN approach, a database is searched for identifying items most similar to a given query item, with similarity determined by a defined distance function. Each item in the database is associated with a label (class), and the primary goal of the algorithm is to determine the class of a new state [25].

kNN relies on two user-defined hyperparameters: the number of nearest neighbors (k) and the choice of distance functions. The parameter k indicates the number of nearest neighbor samples considered to determine the class of the query sample within the dataset. Distance functions play an important role in measuring the distance between samples, significantly affecting the performance of the kNN classifier. Commonly used distance metrics include Euclidean distance and Manhattan distance in the kNN technique. Optimizing these hyperparameters is essential for achieving optimal performance across different datasets.

2.1.2.1. Euclidean Distance

The Euclidean distance, widely used in machine learning, shows the distance between points on a straight line. Computing the distance between two points is based on the Pythagorean theorem. Euclidean distance is determined by taking the square root of the sum of squared difference between two vectors. Mathematically, the Euclidean distance is defined as:

$$d_{Euclidean}(x_1, x_2) = \left(\sum_{k=1}^{n}(x_{1i} - x_{2i})^2\right)^{\frac{1}{2}} \tag{2.7}$$

where $d_{Euclidean}$ is the Euclidean distance between the two samples, x_1 and x_2 represent the two samples whose distances are to be calculated and i is the features in these samples, and n is equal to the number of features in the dataset.

2.1.2.2. Manhattan Distance

The Manhattan distance, also known as city block distance or taxi-cab distance, expresses the absolute difference of the coordinates of x and y objects in D space [26]. Mathematically, Manhattan distance is given by

$$d_{Manhattan}(x_1, x_2) = \sum_{k=1}^{n}|x_{1i} - x_{2i}| \tag{2.8}$$

A visualization of the Euclidean and Manhattan distances employed for determining nearest neighbors for 2D data is given in Figure 2.

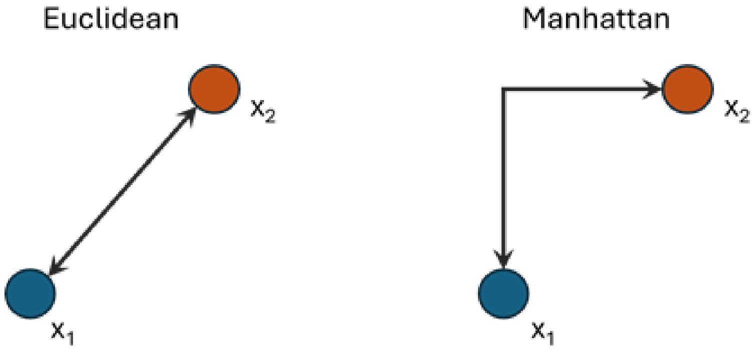


Figure 2. Distance computation with Euclidean and Manhattan metrics in 2D data

2.1.3. Decision Tree (DT)

The DT method is the process of classifying each observation, starting from the root node to the leaves, assigning a “Yes” or “No” outcome based on specific situation. A splitting condition is applied to each node to produce homogeneous subsets. The best split conditions can be selected this way. Impurity measurement is performed for each split state to select the split condition with the lowest impurity value. Various indices such as Gini index, Information gain, gain ratio and misclassification rate have been proposed in the literature to measure the impurity value of a split condition. This study will specifically investigate the effect of Gini index and information gain on classification [27]. The structure of the decision tree is shown in Figure 3.

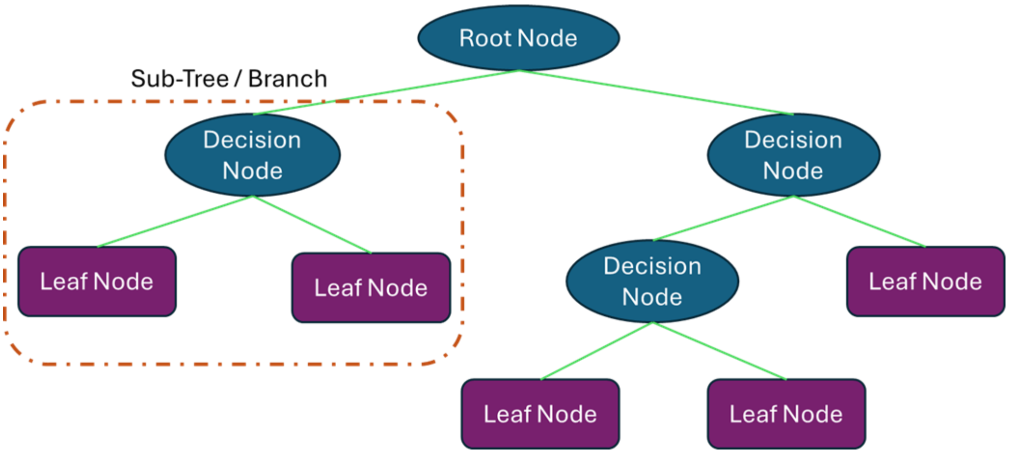


Figure 3. Structural representation of Decision Tree

2.1.3.1. Gini Index

Consider a learning example denoted as $L = \{(x_1, c_1), (x_2, c_2), \cdots, (x_i, c_j)\}$ where $x_1, x_2, \cdots, x_i$ represents an observation vector, and $c_1, c_2, \cdots, c_j$ are the class labels. x_i is a vector of input variables. The division conditions depend on one of these variables. Let p_i denote the probability that a random beam belongs to class c_i [28]. After p_i can be measured as (2.9):

$$p_i = \frac{C_i}{L} \tag{2.9}$$

The Gini index serves as a metric to demonstrate the purity of a class. In the process of determining purity, dataset is initially divided with respect to a specific characteristic, and the resulting clusters’ purity improves with a well-executed division. If we denote *L* as a dataset with *J* different class labels, the Gini index is computed by (2.10) [29].

$$Gini(L) = 1 - \sum_{i=1}^{j} p_i^2 \tag{2.10}$$

where p_i is relative frequency if class *i* in *L*. If the dataset is divided by attribute *A* into two subsets L_1 and L_2 with dimensions N_1 and N_2 respectively, Gini is computed as (2.11):

$$Gini_A(L) = \frac{N_1}{N} Gini(L_1) + \frac{N_2}{N} Gini(L_2) \tag{2.11}$$

Reduction in impurity computed as (2.12):

$$\Delta Gini(A) = Gini(L) - Gini_A(L) \tag{2.12}$$

2.1.3.2. Entropy

Information gain is based on entropy, a measure of impurity or randomness in a dataset [30]. Homogeneous subsets within a dataset imply no impurity or randomness. If all measurements in the subsets belong to a one class, the entropy value for the dataset will be 0. The computation involves summing the probability of each label and the log probability of the same label, denoted by (2.13):

$$Entropy(L) = -\sum_{i=1}^{j} p_i \log_2(p_i) \tag{2.13}$$

For a dataset with one class label, p_i will be 1 and $\log_2(p_i)$ is 0. If the dataset is homogeneous, entropy is zero. As uncertainty, impurity and mixing ratio increase, the entropy value also increases [31].

2.1.4. Random forest (RF)

RF algorithm aims to increase the classification accuracy by employing multiple decision trees during the classification process. It incorporates several hyperparameters that control the structure of each tree for example, the minimum node size for node division. In addition, it influences the overall structure and size of the forest including the number of trees, and the level of randomness presented in the model [32].

The number of trees in a forest is a parameter that cannot be adjusted in the classical sense but needs to be set high enough. For certain error metrics, out-of-bag error curves (somewhat) are observed from time to time, increasing with the number of trees. The rate of convergence, and therefore the number of trees required to achieve optimum performance, depends on the characteristics of the dataset.

Oshiro et al. and Probst and Boulesteix using multiple datasets empirically show that the most remarkable performance improvement usually occurs with the initial 100 trees [33, 34]. The convergence speed of RF relies on both the dataset's characteristics and the hyperparameters. A lower sample size, higher node size values, and smaller input values lead to fewer correlated trees. Since these trees are different from each other, they are expected to provide better predictions. Based on this fact, it is thought that convergence can be achieved by using more trees [32]. Moreover, the level of tree depth also affects the computation rate. The complexity of each decision tree is controlled by the maximum tree depth. The computational cost increases with increasing tree depth. The optimal depth depends on other forest parameters and data characteristics. Error can be reduced by ensuring appropriate depth. High depth may lead to overfitting. Increasing the depth too much reduces the prediction stability. The highest stability will be achieved by using shallow trees, but too much shallowness renders the model inadequate [35].

In the experiment, the training data was divided into random subsets and results were obtained for each subset. Subsequently, all the results were combined, and a majority voting approach was applied. Eventually, a result was obtained. The process of the random forest is depicted in Figure 4.

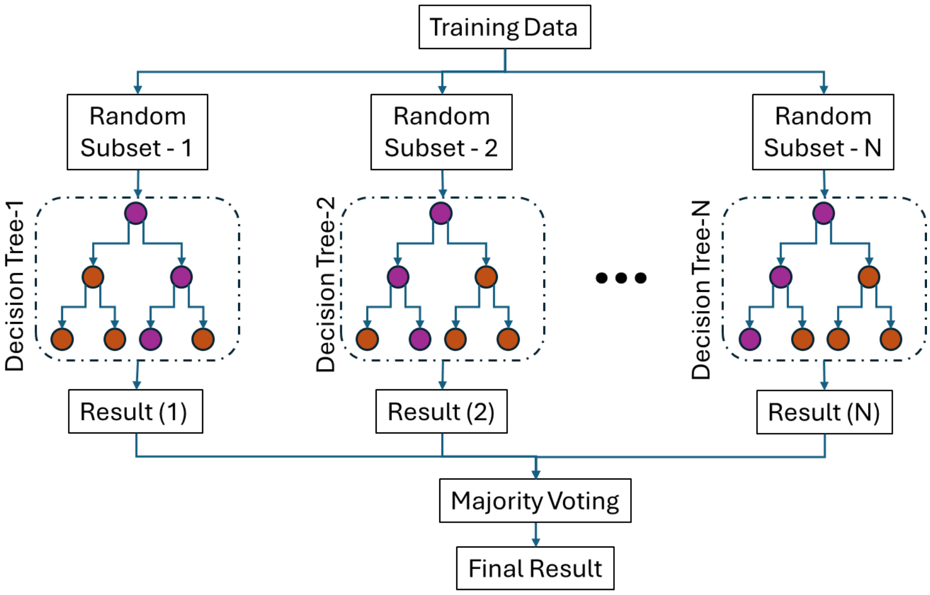


Figure 4. Graphical representation of Random Forest

2.1.5. Support Vector Machine (SVM)

The SVM are a non-parametric statistical learning method that operates in a supervised manner without making any assumptions about the data distribution. Initially introduced by Vapnik and his group in the 1970s, SVM operates on a kernel-based algorithm, typically defining a singular boundary between classes. It divides a dataset into predefined classes using training data. SVM determines the optimal hyperplane to be able to make a separate number of classifications. To maximize separation, it uses support vectors that are closest to the decision boundary of the training sample [36].

The linear separable SVM separates the training sample dataset samples of different categories in sample space where a training sample set is M with the appropriate dividing hyperplane.

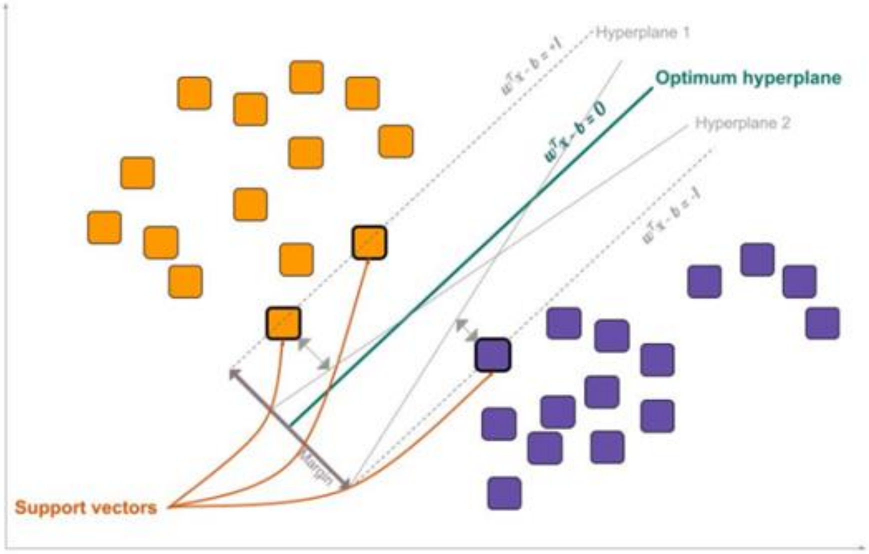


Figure 5. Linear SVM model

In the sample space, the following (2.14) is used to calculate the maximum interval, which expresses the division of the hyperplane.

$$\boldsymbol{w}^T x + b = 0 \tag{2.14}$$

where, **w** is a weight vector and b is a bias term. If the hyperplane can classify the samples correctly [37], then it verifies the following (2.15)

$$\begin{aligned} w^t x_i + b \geq 1, \quad & y = 1 \\ w^t x_i + b \leq -1, \quad & y = -1 \end{aligned} \tag{2.15}$$

The aim of the SVM model is to find optimally the **w** and b values. Thus, the hyperplane separates the data and maximizes the margin $\frac{2}{\|w\|}$. The linear SVM model is shown in Figure 5.

3. Results and Discussion

3.1. Performance Metrics

Performance metrics are utilized to assess the performance of machine learning methods. Studies in the literature have generally focused on binary classification problems in which there are only two classes. In binary classification problems 2×2 confusion matrix is generated but as the number of classes increases the dimension of the confusion matrix changes the $N \times N$ where N is the number of classes in the dataset. Consequently, the performance metrics derived from the confusion matrix undergo adjustments. The confusion matrix for N number of classes in shown in Table 1. i^{th} row j^{th} column in the confusion matrix corresponds to the element of C_{ij}. C_{ij} denotes the number of instances classified as C_j by the machine learning model from instances whose true class is C_i. The confusion matrix for multi-class classification is shown in Table 1.

Table 1. Confusion matrix for multi-class classification

		Predicted Class		
		C_1	$\cdots C_j \cdots$	C_N
Actual Class	C_1	C_{11}	C_{1j}	C_{1N}
	$\vdots$		$\vdots$	
	C_i	C_{i1}	$\cdots C_{ij} \cdots$	C_{iN}
	$\vdots$		$\vdots$	
	C_N	C_{N1}	C_{Nj}	C_{NN}

- True Positive (TP): The actual class is C_i, and it is predicted as C_j where $i = j$.
- False Positive (FP): value for a i^{th} class is the sum of the values in j^{th} column except C_{ij} where $i = j$
- False Negative (FN): value for a i^{th} class is the sum of the values in i^{th} row except C_{ij} where $i = j$
- True Negative (TN): equals to the sum of all the values in confusion matrix except for i^{th} row j^{th} column values where $i = j$

Performance metrics derived from the confusion matrix, provide valuable insights into the effectiveness of machine learning models in multi-class classification scenarios. Some commonly used metrics for multi-class classification include accuracy, micro average precision (MAP), micro average recall (MAR) and macro average F1-score.

Accuracy: measures the correct classification performance of the model. It is calculated as the ratio of the sum of all correctly classified samples to the number of samples in the dataset. It is defined as

$$Accuracy = \frac{\sum_{i=1}^{N} C_{ii}}{\sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}} \tag{3.1}$$

Macro average precision (MAP): MAP equals the average of precisions for each class in the dataset. It is defined as

$$MAP = \frac{1}{N}\sum_{i=1}^{N}\frac{C_{ii}}{C_{.i}}$$

where

$$C_{.i} = \sum_{j=1}^{N} C_{ji}, \quad \forall i \in \{1, \cdots, N\} \tag{3.2}$$

Macro average recall (MAR): The MAR is computed by averaging the recalls for each class in the dataset. It is given as

$$MAR = \frac{1}{N}\sum_{i=1}^{N}\frac{C_{ii}}{C_{i.}}$$

where

$$C_{i.} = \sum_{j=1}^{N} C_{ij}, \quad \forall i \in \{1, \cdots, N\} \tag{3.3}$$

Macro F1-Score combines the MAP, MAR metrics into a single performance metric, and is equal to their harmonic mean. Macro F1-score combines is evaluated by

$$MacroF1 - Score = \frac{1}{N}\sum_{i=1}^{N}\frac{2\frac{C_{ii}}{C_{.i}}\frac{C_{ii}}{C_{i.}}}{\frac{C_{ii}}{C_{.i}} + \frac{C_{ii}}{C_{i.}}} \tag{3.4}$$

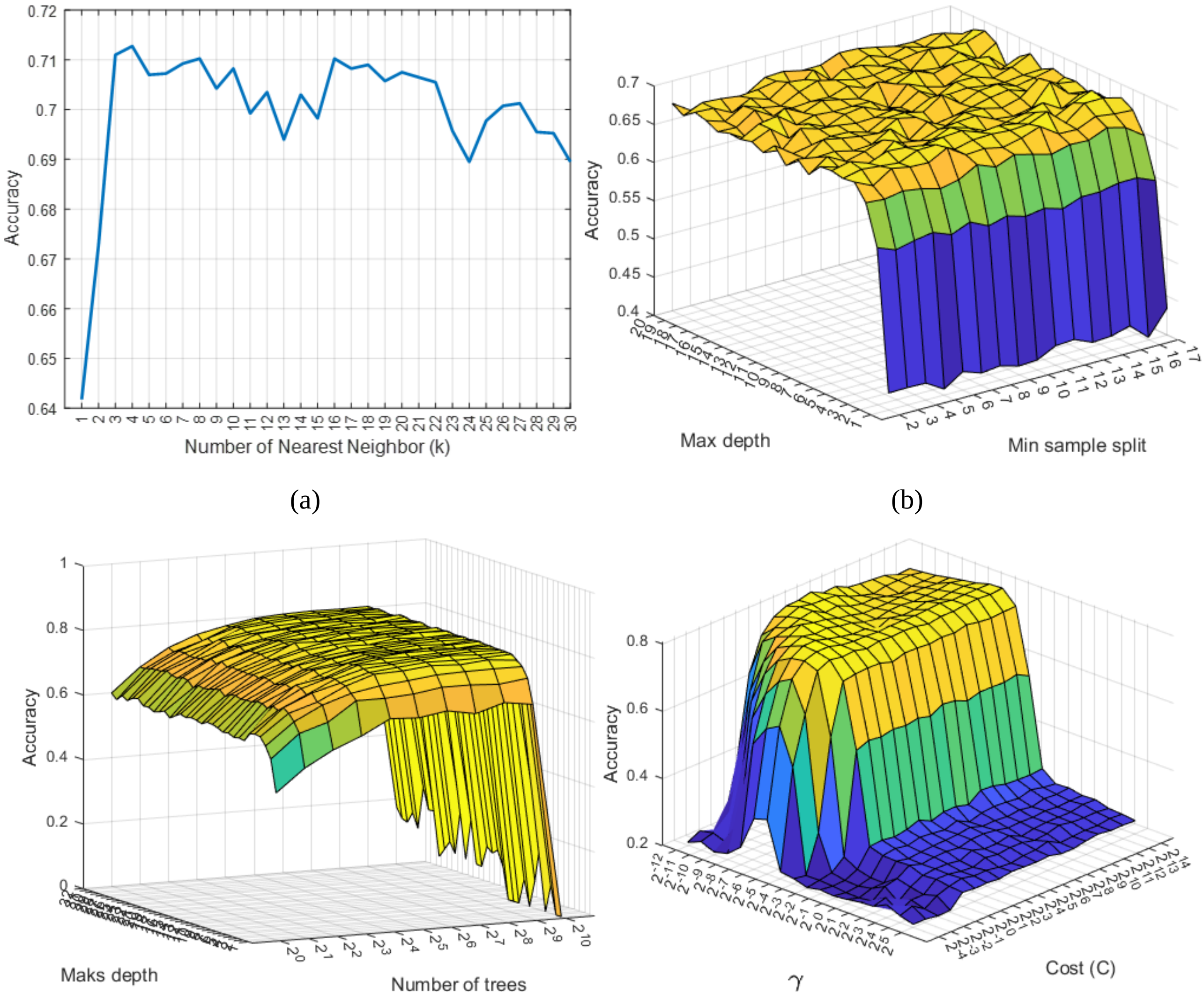
3.2. Experimental Results

Emotions in Turkish music is predicted by four different machine learning methods: kNN, DT, RF, and SVM. Because these methods take user-defined hyperparameters, their performance is affected by hyperparameter settings. To optimize the performance and achieve the highest accuracy in these algorithms, a grid search approach is employed to identify the most proper hyperparameter values. The specific hyperparameters and their respective ranges for each algorithm are detailed in Table 2. Furthermore, *k*-fold cross validation technique is performed instead of a simple train-test split to measure the performance of the benchmarked methods. This approach ensures that all samples in the dataset are used for both training and testing the model, leading to more robust and reliable results.

In the conducted experiments, the *k* value of the *k*-fold cross method was chosen as 5, and the process is iterated 10 times. The average 5-fold cross validation results of the benchmarked methods with the specified hyperparameters are visually presented in Figure 6 (a)-(d).

Table 2. Benchmarked classification methods, method specific hyperparameters and range of values

Method	Hyperparameter	Value
k Nearest Neighbor	Number of nearest neighbor	3, 4, 5, … , 30
Decision Tree	Criterion max_depth min_samples_split	“gini”, “entropy” 1, 2, … , 20 2, 3, … , 17
Random Forest	Criterion max_depth number of trees	“gini”, “entropy” 1, 2, … , 32 2^0, 2^1, … , 2^1
Support Vector Machines	Kernel Cost Gamma Degree	“linear”, “poly”, “rbf”, “sigmoid” 2^{-4}, 2^{-13}, … , 2^{14} 2^{-12}, 2^{-9}, … , 2^5 1, 2, … , 19



(a)

(b)

(c)

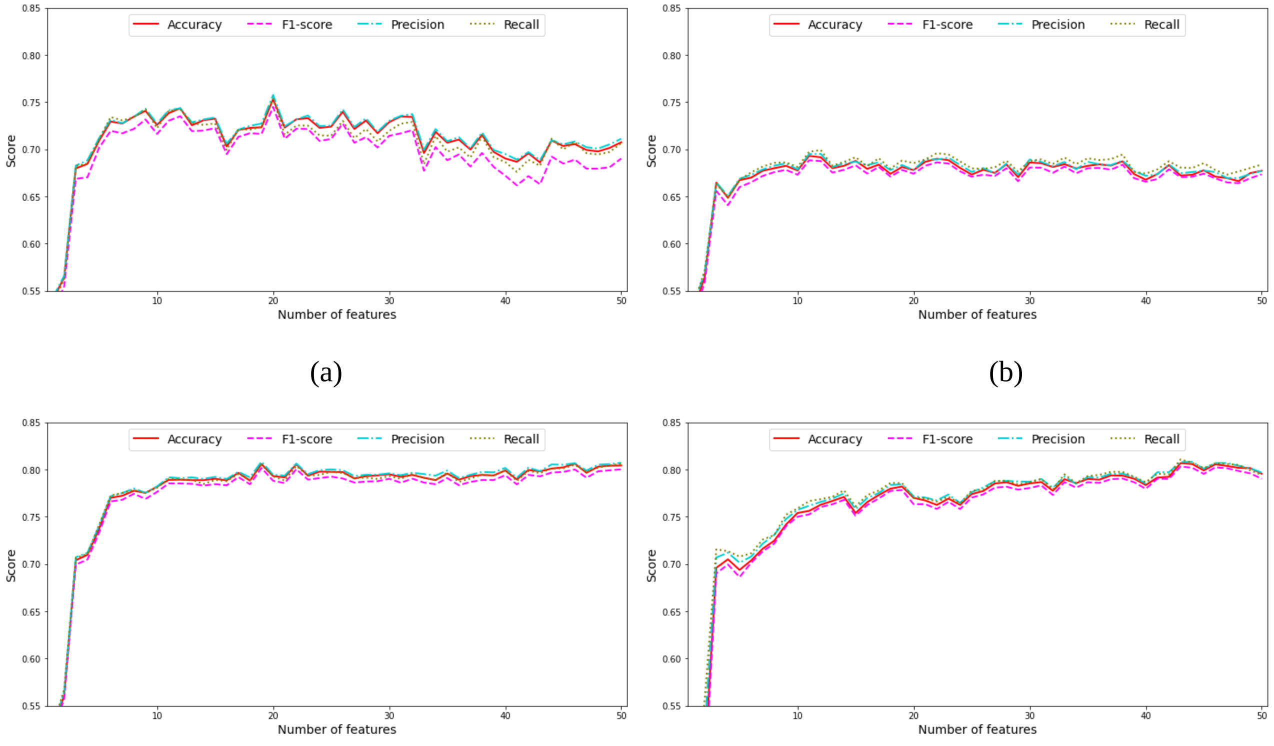
(d)

Figure 6. Accuracy results of the benchmarked methods (a) kNN, (b) Decision Tree, (c) Random Forest, (d) Support Vector Machines

In Figure 6 (a), the results of the kNN algorithm for varying *k* values are presented. The kNN achieves maximum accuracy 71.48% with *k* = *4*. In cases where the *k* value is greater than 4, the results exhibit fluctuations with an example being $k = 16$, where the accuracy value 71.03% is quite close to the highest value. The results are very similar except for $k = 1$. The results of the DT algorithm concerning the *min_sample_split* and *max_depth* hyperparameters are displayed in Figure 6 (b). The effect of the *min_sample_split* hyperparameter is notably limited, while the results are significantly affected when the *max_depth* hyperparameter is less than 3. The performance of the RF method, considering the *max_depth* and the number of trees hyperparameters, is illustrated in Figure 6 (c). “Gini” and “entropy”, which are the splitting node criterion for the RF method, are examined, and since Gini outperformed, its results are shown. Similar observations to the results in DT can be made for the RF method, as changing the number of trees hyperparameter has little effect on the results. Accuracy is significantly reduced when the *max_depth* hyperparameter is greater than 2^{10}. At values where the *max_depth* hyperparameter values begin to drop from 2^4, the accuracy of the RF decreases. In SVM, four different kernels are evaluated with the specified hyperparameter setting, but only the radial-based kernel (RBF) results are shown in Figure 6 (d), as it yields the highest accuracy. The accuracy of SVM with RBF kernel decreases when the gamma value is greater than 2^{-4} and a decrease is also observed when the C value falls below 2^{-1}.

3.2.1. Feature Selection

To identify the most prominent features the MI feature selection method is applied to the Turkish music emotion dataset. The subsets of the dataset are formed based on the features ranked from the highest to the lowest in terms of prominence. Benchmark methods are then applied to these subsets to determine the best performing method using the least number of features. The hyperparameters of the benchmarked methods with the highest accuracy scores were determined in the previous subsection. The performance of these methods is assessed using four different performance metrics: accuracy, precision, recall, and f1-score is illustrated in Figure 7 (a)-(d).



(a)

(b)

(c)

(d)

Figure 7. Performance results of the compared methods depending on the increasing number of features

The results of the kNN method are shown in Figure 7 (a). For kNN, the highest accuracy (75.33%) is obtained when the number of features is equal to 19. However, with 12 features, an accuracy (74.35%) score is attained. Comparing the results using features 4 through 51, the variation between the highest (0.7533) and lowest (0.6845) results is 10.05%. Figure 7 (b) illustrates the performance results of the DT method. The optimal accuracy in DT (0.6930) is obtained with 11 features. When considering features between 5 and 50, the accuracy scores of DT vary between 0.6663 and 0.6930. Figure 7 (c) presents the results of the RF method. As can be seen in the Figure 7 (c), the performance using fewer features in the RF method was not superior to the performance using all features. Considering the trade-off between the number of features and accuracy, the optimum number of features will be 19. This is because the accuracy when using the 19 features with the highest MI score (0.8058) is very close to the accuracy when using all features in the dataset (0.8098). Additionally, the accuracy of models created with more than 15 subsets of features consistently remains above 0.7900. Figure 7 (d) depicts the result of the SVM method. Although SVM results may fluctuate depending on the number of features, there is a general tendency for increased performance as the number of features rises. The highest accuracy (0.8068) in SVM is achieved with 43 features.

The accuracy results of the comparative methods with the specified hyperparameters, using both all features in the dataset and a subset of the features, are given in Table 3. The Table also shows in parentheses the number of features used for the corresponding accuracy values. Feature selection leads to enhanced performance in the kNN and SVM methods, while DT and RF methods experience a decline in performance. Notably, in kNN method, a performance increase over 5% is observed when utilizing only 40% of features. Conversely, in the DT method, there is a minimal loss of less than 0.5% in accuracy with the using of approximately 20% of the features. Similar observations can be made for the RF method, where a loss in performance of less than 0.5% is maintained while using less than 40% of the features. The highest accuracy value (0.8068) was achieved when the MI method is combined with the SVM method; but this success needs the utilization of more than 80% of the features. While the accuracy of SVM+MI (0.8068) slightly surpasses RF+MI (0.8058) by only 0.001, RF+MI uses less than 40% fewer features compared to SVM+MI. Considering the experimental results, RF+MI is the best choice when trying to optimize both accuracy and memory usage efficiency.

Table 3. Accuracy results of the benchmarked method with/ without feature selection

Methods	Hyperparameters	All features	Mutual Information
k Nearest Neighbor	k = 4	0.7128	0.7533 (20)

Decision Tree	Criterion = "gini"
	min_samp_split = 15
	maks_depth = 8
		0.6958	0.6930
			(11)

Random Forest	Criterion = "gini"
	maks_depth = 17
	number of trees = 2^8
		0.8098	0.8058
			(19)

Support Vector Machines	Kernel = "rbf"
	C = 2^2
	$\gamma = 2^{-9}$
		0.8065	0.8068
			(43)

4. Conclusion

In this study, four distinct machine learning algorithms – kNN, DT, RF, and SVM – were applied to determine the emotions from the Turkish music. A rigorous examination of method-specific hyperparameters was conducted, revealing that the RF method, utilizing hyperparameters such as node *splitting criterion = "Gini"*, *number of trees = 28*, and *max_depth = 10*, achieved the highest accuracy (0.8118) when using all features in the dataset, which includes 50 features.

To identify the most descriptive features in the dataset, the MI feature selection algorithm was employed. Comparative analyses were then conducted on subsets of the dataset, using the hyperparameters obtaining the highest accuracy with all features. In this way, it is aimed to achieve a success rate comparable to or higher than the result when all features were used, while using fewer features. The adoption of fewer features not only led to decreased memory consumption but also resulted in processing time.

The SVM method, when coupled with MI feature selection, gave the best results despite utilizing a large number of features. Conversely using the RF method with feature selection, yielded very close results, but with significantly fewer features. As a result, RF+MI emerges as the optimal choice considering both memory usage efficiency and accuracy. This is evidenced by a marginal drop-in success rate of 0.5% compared to using all features and using 60% fewer features.

Future studies may investigate model interpretation methods to elucidate the interpretability of these models. Additionally, different feature selection techniques can be explored to identify the most salient features in the dataset.

Author Contributions

The first author performed supervision, literature review, investigation, verification, arrangement. The first, second, and third authors devised the main conceptual ideas and developed the theoretical framework. The second author performed the experiment and statistical analyses. The third author reviewed and edited the paper. All authors read and approved the final version of the paper.

Conflicts of Interest

All the authors declare no conflict of interest.

References

[1] C. Ji, J. Zhao, Q. Nie, S. Wang, *The role and outcomes of music therapy during pregnancy: a systematic review of randomized controlled trials*, Journal of Psychosomatic Obstetrics & Gynecology 45 (1) (2024) 2291635 10 pages.

[2] G. Leslie, A. Ghandeharioun, D. Zhou, R. W. Picard, *Engineering music to slow breathing and invite relaxed physiology*, 8th international conference on Affective Computing and Intelligent Interaction (ACII), Cambridge, 2019, pp.1–7.

[3] M. Umbrello, T. Sorrenti, G. Mistraletti, P. Formenti, D. Chiumello, S. Terzoni, *Music therapy reduces stress and anxiety in critically ill patients: a systematic review of randomized clinical trials*, Minerva Anestesiologica 85 (8) (2019) 886–898.

[4] P. Vuust, O. A. Heggli, K. J. Friston, M. L. Kringelbach, *Music in the brain*, Nature Reviews Neuroscience 23 (5) (2022) 287–305.

[5] B. M. O. Shimada, M. A. Cabral, V. O. Silva, G. C. Vagetti, *Interventions among pregnant women in the field of music therapy: A systematic review*, Revista Brasileira de Ginecologia e Obstetrícia/RBGO Gynecology and Obstetrics 43 (05) (2021) 403–413.

[6] D. Han, Y. Kong, J. Han, G. Wang, *A survey of music emotion recognition*, Frontiers of Computer Science 16 (6) (2022) 166335 11 pages.

[7] T. Fritz, S. Jentschke, N. Gosselin, D. Sammler, I. Peretz, R. Turner, A. D. Friederici, S. Koelsch, *Universal recognition of three basic emotions in music*, Current Biology 19 (7) (2009) 573–576.

[8] A. Mahadik, S. Milgir, J. Patel, V. B. Jagan, V. Kavathekar, *Mood based music recommendation system*, International Journal of Engineering Research & Technology (IJERT) 10 (6) (2021) 553–559.

[9] A. O. Durahim, A. C. Setirek, B. B. Özel, H. Kebapci, *Music emotion classification for Turkish songs using lyrics*, Pamukkale University Journal of Engineering Sciences 24 (2) (2018) 292–301.

[10] M. B. Er, I. B. Aydilek, *Music emotion recognition by using chroma spectrogram and deep visual features*, International Journal of Computational Intelligence Systems 12 (2) (2019) 1622–1634.

[11] D. Chaudhary, N. P. Singh, S. Singh, *Development of music emotion classification system using convolution neural network*, International Journal of Speech Technology 24 (2021) 571–580.

[12] M. T. Quasim, E. H. Alkhammash, M. A. Khan, M. Hadjouni, *RETRACTED ARTICLE: Emotion-based music recommendation and classification using machine learning with IoT Framework*, Soft Computing 25 (18) (2021) 12249–12260.

[13] J. H. Su, T. P. Hong, Y. H. Hsieh, S. M. Li, *Effective music emotion recognition by segment-based progressive learning*, 2020 IEEE International Conference on Systems, Man, and Cybernetics, Toronto, 2020, pp. 3072–3076.

[14] A. G. Pandrea, J. S. Gómez Cañón, H. Boyer, *Cross-dataset music emotion recognition: an end-to-end approach*, in: J. Cumming, J. H. Lee, B. McFee, M. Schedl, J. Devaney, C. McKey, E. Zangerle, T de Reuse (Eds.), 21st International Society for Music Information Retrieval Conference, Québec, 2020, 2 pages.

[15] T. Ciborowski, S. Reginis, D. Weber, A. Kurowski, B. Kostek, *Classifying emotions in film music—A deep learning approach*, Electronics 10 (23) (2021) 2955 22 pages.

[16] Z. Huang, S. Ji, Z. Hu, C. Cai, J. Luo, X. Yang, *ADFF: Attention based deep feature fusion approach for music emotion recognition* (2022) 5 pages, https://doi.org/10.48550/arXiv.2204.05649.

[17] K. Zhang, X. Wu, R. Tang, Q. Huang, C. Yang, H. Zhang, *The JinYue database for huqin music emotion, scene and imagery recognition*, in: L. O' Conner (Ed.), In 2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR), Tokyo, 2021, pp. 314–319.

[18] D. Moldovan, *Binary horse optimization algorithm for feature selection*, Algorithms 15 (5) (2022) 156 27 pages.

[19] W. Feng, J. Gou, Z. Fan, X. Chen, *An ensemble machine learning approach for classification tasks using feature generation*, Connection Science 35 (1) (2023) 2231168 23 pages.

[20] M. Çakır, A. Degirmenci, O. Karal, *Exploring the behavioural factors of cervical cancer using ANOVA and machine learning techniques*, in: J. Filipe, A. Ghosh, R. O. Prates, B. Horizonte, L. Zhou (Eds.), International Conference on Science, Engineering Management and Information Technology, Ankara, 2022, pp. 249–260.

[21] H. Li, B. Wan, D. Chu, R. Wang, G. Ma, J. Fu, Z. Xiao, *Progressive geological modeling and uncertainty analysis using machine learning*, ISPRS International Journal of Geo-Information 12 (3) (2023) 97 19 pages.

[22] J. Gonzalez-Lopez, S. Ventura, A. Cano, *Distributed multi-label feature selection using individual mutual information measures*, Knowledge-Based Systems 188 (2020) 105052 13 pages.

[23] P. Zhang, G. Liu, J. Song, *MFSJMI: Multi-label feature selection considering join mutual information and interaction weight*, Pattern Recognition 138 (2023) 109378.

[24] J. R. Vergara, P. A. Estévez, *A review of feature selection methods based on mutual information*, Neural Computing and Applications 24 (2014) 175–186.

[25] A. Degirmenci, O. Karal, *iMCOD: Incremental multi-class outlier detection model in data streams*,

Knowledge-Based Systems 258 (2022) 109950.

[26] R. Suwanda, Z. Syahputra, E. M. Zamzami, *Analysis of Euclidean distance and Manhattan distance in the K-means algorithm for variations number of centroid K*, Journal of Physics: Conference Series 1566 (1) (2020) 012058 6 pages.

[27] A. N. Karaoglu, H. Caglar, A. Degirmenci, O. Karal, *Performance improvement with decision tree in predicting heart failure*, 6th International Conference on Computer Science and Engineering (UBMK), Ankara, 2021, 781–784.

[28] S. Tangirala, *Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm*, International Journal of Advanced Computer Science and Applications 11 (2) (2020) 612–619.

[29] M. A. Bouke, A. Abdullah, S. H. ALshatebi, M. T. Abdullah, H. El Atigh, *An intelligent DDoS attack detection tree-based model using Gini index feature selection method,* Microprocessors and Microsystems 98 (2023) 104823.

[30] S. Yao, Y. Wu, F. Akter, *An introduction to artificial intelligence and machine learning,* in: F. Akter, N. Emptage, F. Engert, M. S. Berger (Eds.), Neuroscience for Neurosurgeons, Cambridge University Press, 2023, Ch. 9, pp.146–157.

[31] R. Mirzaeian, R. Nopour, Z. Asghari Varzaneh, M. Shafiee, M. Shanbehzadeh, H. Kazemi-Arpanahi, *Which are best for successful aging prediction? Bagging, boosting, or simple machine learning algorithms?,* Biomedical Engineering Online 22 (1) (2023) 85 25 pages.

[32] M. Apaydın, M. Yumuş, A. Değirmenci, Ö. Karal, *Evaluation of air temperature with machine learning regression methods using Seoul City meteorological data*, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi 28 (5) (2022) 737–747.

[33] T. M. Oshiro, P. S. Perez, J. A. Baranauskas, *How many trees in a random forest?*, in: P. Perner (Ed.), Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, Berlin, 2012, 154–168.

[34] P. Probst, A. L. Boulesteix, *To tune or not to tune the number of trees in random forest*, The Journal of Machine Learning Research 18 (1) (2017) 6673–6690.

[35] C. B. Liu, B. P. Chamberlain, D. A. Little, Â. Cardoso, *Generalising random forest parameter optimisation to include stability and cost,* in: M. Ceci, J. Hollmén, L. Todorovski, C. Vens, S. Džeroski (Eds.), Machine Learning and Knowledge Discovery in Databases: European Conference, Skopje, 2017, 102–113.

[36] M. Muttaqi, A. Degirmenci, O. Karal, *US accent recognition using machine learning methods,* Innovations in Intelligent Systems and Applications Conference (ASYU), Antalya, 2022, 1–6.

[37] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, S. Homayouni, *Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review,* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13 (2020) 6308–6325.