# Expressive Music Performance Modelling

**Andreas Neocleous**

MASTER THESIS UPF / 2010
Master in Sound and Music Computing

Master thesis supervisor: Rafael Ramirez
Department of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona

# Acknowledgements

I would like to thank my advisor Prof Rafael Ramirez for his consistent and valuable support during the process of research and preparation of this thesis. I would also like to thank Prof Xavier Serra for his support and the opportunity he gave me to be part of the music technology group. I am also grateful to Esteban Maestre, Alfonso Perez and Panos Papiotis for their help, valuable comments and suggestions. Finally, I would like to thank my family for their endless support.

# Abstract

Machine learning approaches to modelling emotions in music performances were investigated and presented in this thesis. In particular, we investigated how professional musicians encode emotions, such as happiness, sadness, anger, fear and sweetness, in violin and saxophone audio performances. Suitable melodic description features were extracted from audio recordings. Following that, we applied various machine learning techniques for training expressive performance models. A model was trained for each emotion considered. Finally, new expressive performances were synthesized from inexpressive melody descriptions (i.e. music scores) using the induced models and the result was perceptually evaluated by asking a number of people to listen, compare and evaluate to the computer generated performances. Several machine learning techniques for inducing the expressive models were systematically explored and we present the results.

# Index

# List of figures

page

# List of tables

# 1. Introduction

## 1.1 Motivation

There are a large number of emotions that people experience in their everyday life. Many thinkers, for many centuries, tried to understand where these emotions arise, what purposes they serve, and how and why we have distinctive feelings.

In music, the composers can use several techniques in order to generate emotions that may be felt by listeners. For instance they may use minor scales when they want to create a sad melody and major scales when they want to crate a happy melody. On the other hand, performers use other techniques to cause different emotions. For example, a diminished seventh chord with rapid tremolo can evoke suspense. A melody can be funny and cause laugh if the musician plays a sequence of notes with fast changes and with a big distance in frequency between them. Also, a combination between changes in timbre, duration and dynamics may create different emotions. A melody with hard attacks, tough timbre and short durations could give the sensation of an angry melody. In contrast, the same melody with soft attacks, poor timbre, longer durations and unstable dynamics could give the sensation of fear.

Musicians tend to express their emotions while performing, not only by producing different melodies with their instruments, but also by manipulating different sound characteristics such as strength, duration, intonation, timbre, etc. Furthermore, many times they express feelings through the movement of their body, the expressions of their face and other gestures. Each musician uses different ways to express him/herself while performing a musical piece or while just improvising. Thus, the way each musician expresses him/herself is different from the others. The score carries information such as the rhythmic and melodic structure of a certain piece, but as yet there is no notation able to describe precisely the temporal and timbre characteristics of the sound. It is often left to the musician to choose these characteristics in the interpretation of the piece. From the musical point of view, the sound properties that musicians manipulate for conveying expression in their performances are pitch, timing, amplitude, and timbre.

Whenever the information of a musical score is played by a computer, the resulting performance often sounds mechanical and unpleasant. In contrary, a human performer introduces deviations in the timing, dynamics and timbre of the performance, following a procedure that correlates to his/her own experience. This is quite common in instrumental practice. From the measurement of such deviations, general performance patterns and principles can be deduced.

Thus, the motivation for the work that is presented in this thesis is to deal with the measurement and modelling of the expressive deviations introduced by expert musicians while performing musical pieces in an attempt to contribute to the understanding, generation and retrieval of expressive performances.

## 1.2 Objectives

The main goal of this work is to build a computational model which predicts how a musical score should be played in order to give the sensation to a listener that the song has been played by a musician and not by a computer. That means that the model will be able to accurately predict expressive information.

The more specific objectives of the work were to:

1.2.1 Extract suitable audio features from properly generated audio files. These features will symbolically represent the performances.

1.2.2 Apply suitable machine learning techniques on the signals and features, aiming at finding the best possible representational model.

1.2.3 Generate new synthesized scores by using the predictions from the models.

1.2.4 Evaluate the results, by giving suitable questionnaires to knowledgeable persons asking them to distinguish the songs performed by a human, the computer generated, and by score. If the subjects are able to distinguish the difference between the songs generated by the score information and those generated by the prediction information, that means that the predictions are able to add some information which is different than the score.

## 1.3 Research overview/Methodology

The approach to expressive music performance lies at the intersection of the disciplines of Musicology and Artificial Intelligence (in particular machine learning and data mining). The general methodology for the proposed research can be described as follows:

1. Obtain high-quality recordings of performances by human musicians (e.g. violinists) in audio format.
2. Extract a symbolic (machine-readable) representation from the recorded pieces.
3. Encode the music scores of the corresponding pieces in machine readable form. If the score is not available, construct a *virtual* score from the performance.
4. Extract important expressive aspects (e.g. energy variations, timbre manipulation, …) by comparing the recorded scores and the actual performances.
5. Analyze the structure (e.g. meter) of the pieces and represent the scores and their structure in a machine readable format.
6. Develop and apply machine learning techniques that search for expressive patterns among the structural aspects of the pieces and expressive deviations.
7. Perform systematic experiments with different representations, sets of recordings, musical styles and instruments.
8. Analyze the results with the aim of understanding, generating and retrieving expressive performances.

Figure 1.3a illustrates the general research framework of this work.

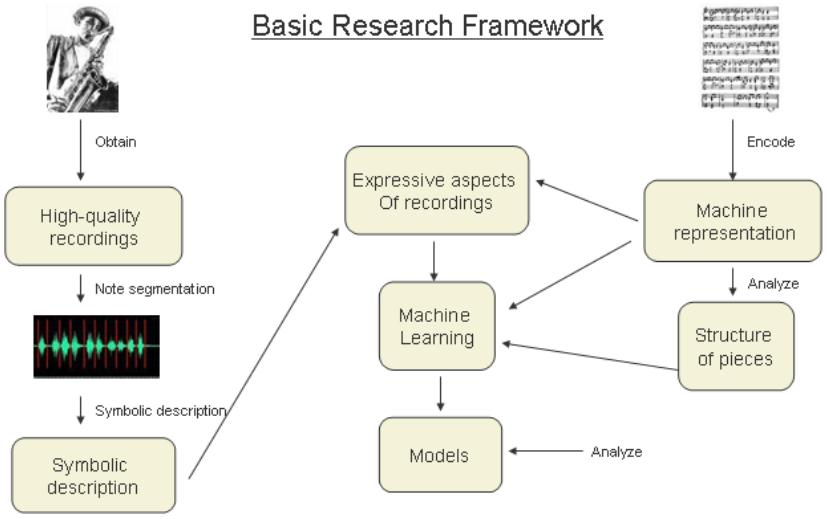


Figure 1.3a Basic research procedure to be followed.

The first step consisted of obtaining high-quality recordings of performances by human musicians in audio format. The performances were recorded in the studio which is located in the campus of the University of Pompeu Fabra. Then a symbolic representation from both the recordings has been extracted. Furthermore, the structure of the pieces and all the information has been analyzed, including the symbolic representation from the audio has been represented in a machine-readable format. After, the machine-readable format has been obtained with all the appropriate information, and machine learning techniques has been developed and applied in order to search for expressive patterns among the structural aspects of the pieces and expressive deviations. Finally, systematic experiments have been performed with different representations and the results were analyzed with the aim of understanding, generating and retrieving expressive performances.

## 1.4 Organization of the thesis

The rest of the thesis is organized as follows. In Chapter 2, the previous work and the state of art will be presented and explained. Following that, in Chapter 3, an introduction to machine learning and the techniques used will be presented. Chapter 4 will present the data and the processing necessary to obtain the suitable audio features. The procedure of extracting the features and the way of computing the onsets for the note segmentation will be explained in detail. In Chapter 5, the methodology for expression identification, and the algorithms and settings used will be described and presented. In 6th Chapter the results will be presented and discussed. Finally, in the last Chapter (7), conclusions will be drawn, and future work will be presented.

# 2. Background

## 2.1 Expressive music performance

Musicians when asked to perform a piece from a written score they make deviations from the score for two main reasons. Firstly, it is very difficult to perform the score as it is and secondly, these deviations can evoke feelings and expressiveness in the performance. Many professional musicians show their character to the performances in a sense that the listeners are able to recognize them from the way they perform. Many famous songs have been played and expressed differently by many different artists. It is an interesting fact that listeners can recognize an artist or a musician even if a song is purely instrumental. For instance in jazz, there are a lot of songs that were played by different famous saxophonists, each putting his own style and expression, conveying different feelings to the listeners. The differences between these are mainly in the instrumentation, but also to the way that the main musician performs the particular song. What are then the differences from the score and the musician performances that make each of them to be special? Why people often say that a particular song is the best, even though there are tens or maybe hundreds of different covers of the same song?

There are a lot of ways for a musician to express emotions in music. These can be differences in the duration of the notes, the dynamics, the differences in timbre, the articulation, the vibrato and so on. In that sense, if we ask a number of musicians to perform a particular song, each musician will most likely perform the song in a different way.

The deviations from the score that each musician might make, will affect the way the song it sounds. This is due to a number of reasons. The first reason is because no one can really perform all the notes with their actual durations. It is very difficult to control the duration of the notes. It can be very close, but it will never be the actual duration. Furthermore, musicians often make deviations from the written duration just because they want to change the mood of the song or just to give attention to a particular part of the song, or for another reason which is always related to expressivity. One more reason is because of the timbre that musicians can change to their instruments. In many instruments the timbre is flexible and it is up to the musician to choose the sound and the timbre of their instruments. For instance in the brass instruments, the timbre can be controlled by the mouthpiece, the position of the tongue, the pressure of the lips and many others. It is all these deviations that I am trying to capture and model them using machine learning techniques. Once they will be accurately modelled, then predictions of similar deviations can be done from unknown scores and then imitations of the way that famous musicians are performing the music can be done.

## 2.2 The state of the art

Expressive music performance research [1] investigates the manipulation of sound properties in an attempt to understand and recreate expression in performances. Expressive performance modelling and style-based performer identification is an important and extremely challenging computer music research topic. Previous work has addressed expressive music performance using a variety of approaches, e.g. [2, 3, 4, 5]. In the past expressive music performance has been studied in different contexts and using different approaches. The main approaches to expressive performance modelling have been (a) empirical, and (b) the machine-learning-based. An interesting question in expressive performance modelling research is how to use the information encoded in the expressive models for the identification of performers. However, the use of expressive performance models for identifying musicians has received little attention in the past.

### 2.2.1 Empirical expressive performance modelling

The main approaches to manually studying expressive performance are three. The first approach is based on statistical analysis [6], the second in mathematical modelling [7], and the third in analysis-by-synthesis [8]. In all these approaches, it is a person who is responsible for devising a theory or mathematical model which captures different aspects of musical expressive performance. The theory or model is later tested on real performance data in order to determine its accuracy.

A lot of research has been done by the KTH group in order to model and explain symbolic (i.e. MIDI) expressive performances. They developed a program called *Director Musices* [9] system which transforms noted scores into musical performances. It incorporates rules for tempo, dynamic, phrasing, articulation, and intonation, and they operate on performance variables such as tone, inter-onset duration, amplitude, and pitch. The rules are obtained from both theoretical musical knowledge, and experimentally by using an analysis-by-synthesis approach. The user of the program can manipulate rule parameters and control different features of the performance. The computer executes all the technical computations in order to obtain different interpretations of the same piece. The rules are divided into three main classes: (1) *differentiation rules,* which enhance the differences between scale tones; (2) *grouping rules,* which specify what tones belong together; and (3) *ensemble rules,* which synchronize the various voices in an ensemble. Most of the research of the KTH group intents to clarify the expressive features of piano performance e.g. [10, 11, 12].

One of the first attempts to provide a computer system with musical expressiveness is that of Johnson (1992) [13]. Johnson manually developed a rule-based expert system to determine expressive tempo and articulation for Bach's fugues from the Well-Tempered Clavier. The rules were obtained from two expert performers.

Canazza et al. (1997) [14] developed a system to analyze the relationship between the musician's expressive intentions and her performance. The analysis reveals two expressive dimensions, one related to loudness (dynamics), and another one related to timing (rubato).

Dannenberg et al. (1998) [15] investigated the trumpet articulation transformations using (manually generated) rules. They developed a trumpet synthesizer which combines a physical model with an expressive performance model. The performance model generates control information for the physical model using a set of rules manually extracted from the analysis of a collection of performance recordings.

### 2.2.2 Machine-learning-based expressive performance modelling

Previous research addressing expressive music performance using machine learning techniques has included a number of approaches.

Lopez de Mantaras and Arcos (2002) [16] report on SaxEx, a performance system capable of generating expressive solo saxophone performances in Jazz. Their system is based on case-based reasoning, a type of analogical reasoning where problems are solved by reusing the solutions of similar, previously solved problems. In order to generate expressive solo performances, the case-based reasoning system retrieves from a memory containing expressive interpretations, those notes that are similar to the input inexpressive notes. The case memory contains information about metrical strength, note duration, and so on, and uses this information to retrieve the appropriate notes. One limitation of their system is that it is incapable of explaining the predictions it makes and it is unable to handle melody alterations, e.g. ornamentations.

Ramirez et al. (2006) [17] have explored and compared diverse machine learning methods for obtaining expressive music performance models for Jazz saxophone that are capable of both generating expressive performances and explaining the expressive transformations they produce. They propose an expressive performance system based on inductive logic programming which induces a set of first order logic rules that capture expressive transformation both at an inter-note-level (e.g. note duration, loudness) and at an intra-note-level (e.g. note attack, sustain). Based on the theory generated by the set of rules, they implemented a melody synthesis component which generates expressive monophonic output (MIDI or audio) from inexpressive melody MIDI descriptions.

With the exception of the work by Lopez de Mantaras et al. and Ramirez et al., most of the research in expressive performance using machine learning techniques has focused on classical piano music e.g. [3, 18, 19], where often the tempo of the performed pieces is not constant. Thus, these works focus on global tempo and loudness transformations.

Widmer has focused on the task of discovering general rules of expressive classical piano performance from real performance data via inductive machine learning. The performance data used for the study are MIDI recordings of 13 piano sonatas by W.A. Mozart performed by a skilled pianist. In addition to these data, the music score was also coded. The resulting substantial data consists of information about the nominal note onsets, duration, metrical information and annotations. When trained on the data the inductive rule learning algorithm named PLCG [2] discovered a small set of 17 quite simple classification rules [20] that predict a large number of the note-level choices of the pianist. In the recordings, the tempo of the performed piece was not

constant, as it was in our experiments. In fact, the tempo transformations throughout a musical piece were of special interest.

### 2.2.3 Expressive performance modelling for performer identification

The use of expressive performance models (either automatically induced or manually generated) for identifying musicians has received little attention in the past. This is mainly due to two factors: (a) the high complexity of the feature extraction process that is required to characterize expressive performance, and (b) the question of how to use the information provided by an expressive performance model for the task of performance-based performer identification.

Saunders et al. (2004) [21] apply string kernels to the problem of recognizing famous pianists from their playing style. The characteristics of performers playing the same piece are obtained from changes in beat-level tempo and beat-level loudness. From such characteristics, general performance alphabets can be derived, and pianists' performances can then be represented as strings. They apply both kernel partial least squares and Support Vector Machines to this data.

Stamatatos and Widmer (2005) [22] address the problem of identifying the most likely music performer, given a set of performances of the same piece by a number of skilled candidate pianists. They propose a set of very simple features for representing stylistic characteristics of a music performer that relate to a kind of 'average' performance. A database of piano performances of 22 pianists playing two pieces by Frederic Chopin is used. They propose an ensemble of simple classifiers derived by both subsampling the training set and subsampling the input features. Experiments show that the proposed features are able to quantify the differences between music performers.

Grachten and Widmer (2009) [23] apply a machine-learning classifier in order to characterize and identify individual playing style of pianists. The feature they used to train the classifier was the differences of the final ritardandi by different pianists. The data they used were recordings of Chopin's and they were taken from commercial CD's. These recordings are chosen on purpose because they exemplify classical piano music from romantic period which is a genre characterized by the prominent role of expressive interpretation in terms of tempo and dynamics.

Ramirez et al (2007) [24] presents an approach of identifying performers from their playing styles using machine learning techniques. The data used in their investigations are audio recordings of real performances by famous Jazz saxophonists. The note features they used represent both properties of the note itself and aspects of the musical context in which the note appears. Information about the note includes note pitch and note duration, while information about its melodic context includes the relative pitch and duration of the neighbouring notes, as well as the Narmour [25] structures to which the note belongs. In [26] they used recordings of Irish popular music performances in order to model the performances of each performer and then automatically identify which one is the input performance by using the models.

# 3. Machine learning

## 3.1 Introduction

Researchers use machine learning (ML) techniques mainly to manipulate large amounts of data, aiming at extracting useful information that is difficult or impossible to obtain by simple observation or through the use of classical statistical techniques. Thus, by using ML they give a useful meaning to data. More specifically, many times it is very difficult, or even impossible, for a human to manually find similarities in data and categorize them according to available information that is often hidden in many numbers. This is largely due to the huge amount of the data and the fast rate of changes. With ML techniques the data can be effectively categorized according to the information they carry. This can be done by using unsupervised or supervised learning. For instance, we might have a play list of songs and we may want to separate the songs into categories according to the genre. If we want to find an intelligent way to do that, there is a multitude of techniques to achieve this. For instance, one method is to use unsupervised ML and let the algorithm classify the songs according to the information in the input. In that case the input can be some appropriate features that contain clues and have information that may help in the proper classification. Such features could be the rhythm, the instrumentation, and other relevant characteristics that can be informative in the sense of classifying the genre.

ML can also be used in a supervised learning manner. Supervised learning means that the algorithm has both the problem and the solution, and is trying to generalize from such instances. Thus, the algorithm is trying to build a model according to the training data. Usually we feed the algorithm with a lot of examples which have some inputs and one or more outputs. With this technique we can build models for a multitude of systems that we are interested and then the trained ML system will be able to predict the output by using the training model. For example, we can build a model for predicting the temperature by giving as output the values of the temperature for one year and as input information about the day, the season, the humidity and others. This will train the machine and it will be able to predict the temperature of the day we need to predict by giving to the input the data of that day.

## 3.2 Evaluation methods

In machine learning, there are several techniques to evaluate a model. One of the most powerful and most common evaluation tool is the cross validation. In cross validation, three methods may be used. These are the **holdout** method which is the simplest one, the **K-fold** cross validation which is an improved method and the **leave-one-out** cross validation.

The basic idea of evaluating a model is to test a set of data that have been trained with a new, unknown data. The idea of cross validation method is to separate the whole set of the data in two subsets, where one is kept out from the training set in order to be used later as the test set.

The holdout method is separating the data into two subsets. One of them is used to train the model called the training set and the other one is used to test the model called the test set. The test set is used later, to be applied to the trained model in order to predict the output values of the data. The error it makes may be expressed as the mean absolute test set error, which is used to evaluate the model.

The K-fold cross validation is very similar to the holdout method. The main difference is that instead of separating the data into one training set and one test set, it separates the data randomly into k-subsets where it trains the model with the k-1 subsets leaving one subset out for the test. This is done k times and the evaluation is the mean of all the k times. In the experiments of the work presented in my thesis, a 10-fold cross validation has been used which is the most common evaluation method.

The leave-one-out cross validation has the same idea with the k-fold cross validation with the difference that the training set is the whole set of the data minus one point which will be the test for the prediction. This is very expensive to compute.

## 3.3 Machine Learning Algorithms

### Decision trees learning algorithm

Trees are very popular tools for regression and classification. The main idea behind this technique is to build rules for the classification or the regression similar to the structure of a tree. A decision tree can be used to classify an example by starting at the root of the tree and moving through until a leaf node is reached, which provides the classification of the instance. In each node, the classifier is moving through the structure by taking a decision. Usually, the test at a node compares an attribute value with a constant. To classify an unknown instance, it is routed down the tree according to the values of the attributes tested in successive nodes, and when a leaf is reached the instance is classified according to the class assigned to the leaf. To make a decision, the attribute with the highest normalized information gain is used. The splitting procedure stops if all instances in a subset belong to the same class. A good measure for selecting the attribute in the node is called **information gain**. Information gain is itself calculated using a measure called entropy. Given a set S, containing only positive and negative examples of some target concept (a 2-class problem), the entropy of set S relative to a binary classification is defined as:

$$\text{Entropy,} \quad S \equiv -p_p \log_2 p_p - p_n \log_2 p_n \qquad \text{(eq. 3.1)}$$

Where, $p_p$ is the proportion of positive examples in S and $P_n$ is the proportion of negative examples in S. If the target attribute takes on *c* different values, then the entropy of S relative to this *c*-wise classification is defined as

$$\text{Entropy,} \quad S = \sum_{i=1}^{c} -p_i \log_2 p_i \qquad \text{(eq. 3.2)}$$

Where $p_i$ is the proportion of S belonging to class *i*. The information gain of attribute A, relative to a collection of examples, S, is calculated as:

$$\text{Information Gain,} \qquad S, A = S - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} S_v \qquad \text{(eq. 3.3)}$$

Where, *Values(A)* is the set of all possible values for attribute *A*, and $S_v$ is the subset of *S* for which attribute *A* has value *v* (i.e., $S_v = \{s \in S \mid A(s) = v\}$).

The tree algorithms used in the work reported in this thesis are the C4.5 (J48 in Weka) for classification and the M5 Rules for regression. The C4.5 is an algorithm developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy as explained above.

Table 3.1 shows an example of the data used in this thesis work for the classification of the bow direction. Table 3.1a shows the features used for the training while table 3.1b shows the values of each attribute. The last name is the class which is used for

the classifier to learn and eventually to build the model. This is the bow direction and the two classes are Change or No Change. These data were trained by the J48 algorithm using the Weka environment and the tree generated is presented in Figure 3.1.

Table 3.1a. The features used for Table 3.1b

**Note duration**
**Previous duration**
**Next duration**
**Previous interval**
**Next interval**
**Metro strength** (Extremely Low, Low, Medium, High, Extremely High)
**Narmour group 0** (none, d, id, reverse id, ip, reverse ip, ir, reverse ir, p, reverse p, r, reverse r, vp, reverse vp, vr, reverse vr, d2, m)

**Narmour group 1** (none, d, id, reverse id, ip, reverse ip, ir, reverse ir, p, reverse p, r, reverse r, vp, reverse vp, vr, reverse vr, d2, m)

**Narmour group 2** (none, d, id, reverse id, ip, reverse ip, ir, reverse ir, p, reverse p, r, reverse r, vp, reverse vp, vr, reverse vr, d2, m)
**Tempo**
**Bow direction** (NoChange, Change)

Table 3.1b. Example of the trained data for the bow direction

| Note duration | Previous duration | Next duration | Previous duration | Next int | metro | Nargroup _0 | Nargroup _1 | Nargroup _2 | Tempo | Bow direction |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.5 | 0 | -0.25 | 0 | 7 | Extremely High | r | none | none | 2 | NoChange |
| 0.25 | 0.25 | 0 | -7 | -2 | Low | p | r | none | 2 | Change |
| 0.25 | 0 | 0.25 | 2 | -1 | Extremely Low | p | r | none | 2 | Change |
| 0.5 | -0.25 | -0.25 | 1 | -2 | Medium | p | none | none | 2 | NoChange |
| 0.25 | 0.25 | 0 | 2 | -2 | Low | id | p | none | 2 | Change |
| 0.25 | 0 | 0.25 | 2 | 2 | Extremely Low | reverse_vr | id | p | 2 | NoChange |
| 0.5 | -0.25 | 0 | -2 | -7 | High | reverse_vr | id | none | 2 | NoChange |
| 0.5 | 0 | 0 | 7 | 0 | Low | reverse_vr | none | none | 2 | NoChange |
| 0.5 | 0 | -0.25 | 0 | 10 | Low | r | none | none | 2 | NoChange |
| 0.25 | 0.25 | 0 | -10 | -1 | Extremely High | p | r | none | 2 | Change |
| 0.25 | 0 | 0 | 1 | -2 | Extremely Low | p | r | none | 2 | NoChange |
| 0.25 | 0 | 0 | 2 | -2 | Low | p | none | none | 2 | NoChange |
| 0.25 | 0 | 0.25 | 2 | -1 | Extremely Low | reverse_vr | p | none | 2 | NoChange |
| 0.5 | -0.25 | 0 | 1 | 6 | Medium | r | reverse_vr | p | 2 | Change |
| 0.5 | 0 | 0 | -6 | 0 | Low | ip | r | reverse_vr | 2 | NoChange |
| 0.5 | 0 | 0 | 0 | -1 | High | ip | r | none | 2 | Change |
| 0.5 | 0 | 0 | 1 | 0 | Low | ip | none | none | 2 | Change |
| 1 | -0.75 | -0.75 | 1 | 0 | Extremely High | reverse_vr | ip | p | 2 | Change |
| 0.25 | 0.75 | 0 | 0 | -4 | Medium | p | reverse_vr | ip | 2 | Change |
| 0.25 | 0 | 0 | 4 | -1 | Extremely Low | id | p | reverse_vr | 2 | Change |

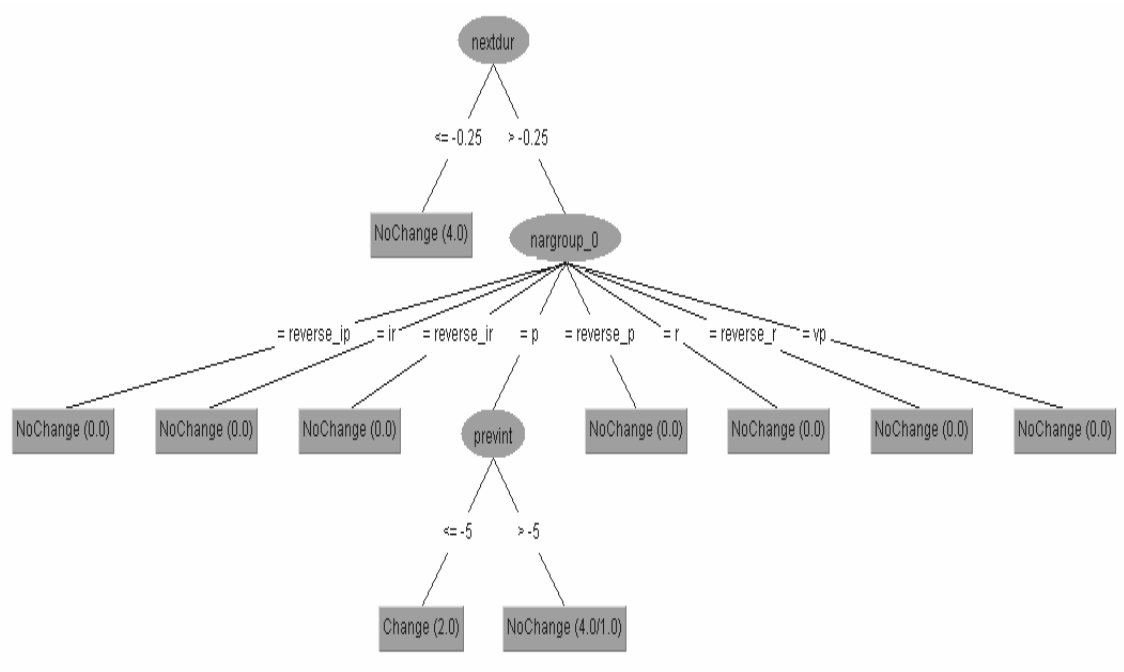


Figure 3.1: Representation of the tree structure of the bow direction model.

## Lazy Methods

Lazy methods store all the training instances in the memory until the time of the classification. There are a number of algorithms that use the technique of lazy methods. In my work, a **k-nearest neighbour** algorithm (KNN) has been used. In KNN, when a new instance has to be classified, it finds the closest instance which is stored in the memory by calculating the Euclidean distance between the unknown instance and the instances used in the training. In one nearest neighbour, the closest instance is only one, thus the class of the unknown instance will be the one that the particular instance belongs. If the algorithm checks for more than one nearest neighbour, then the predicted class of the unknown instance will be the one that has the most training instances. Some times it is better to weigh the data according to the number of the training instances for each class. It is obvious that if a class has much more instances than another, then the probability to appear as a nearest neighbour is high. This is one of the drawbacks of this method. One other drawback of the k-nearest neighbour technique is that in order to predict the classification, it has to have in the memory all the instances. This might cause considerable overhead, if the training data set is very large. For the experiments of this work, 1-K nearest neighbour was chosen as a parameter to the algorithm. Figure 3.2 shows an example of the k-nearest neighbour classification.

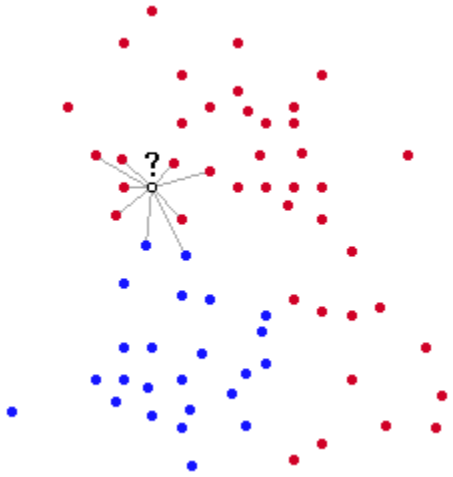


Figure 3.2: Example on how the k-nearest neighbour algorithm classifies a new instance.

## Artificial neural networks

The artificial neural networks (ANN) are a system of interconnected processing elements (usually simple), that presumably work in parallel, in resemblance to biological neural networks [27]. Actually the processing in digital computers is serial, but the simulations are done are very fast that resemble parallel processing. The interconnection is usually dense and structured, and most often displayed in directed graph formalism as shown in Figure 3.3.

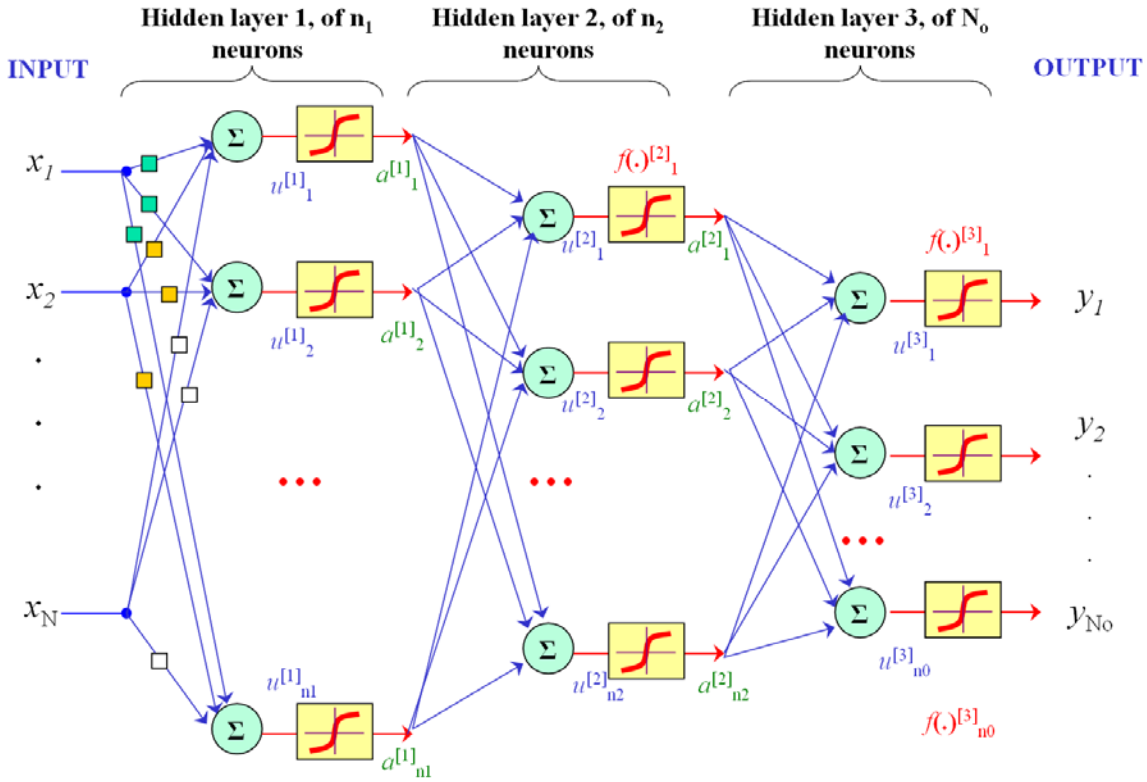


Figure 3.3: A typical feedforward multilayer artificial neural network.

ANNs may be models of biological neural networks (BNN), but most of them are paradigms of models that attempt to produce artificial systems capable of sophisticated, hopefully intelligent computations, similar to those that the human brain routinely performs.

The ANNs are adaptable, through the application of appropriate learning, by using suitable training rules. They usually learn through the application of examples of known inputs-outputs. This is known as supervised training. The most common and one of the most successful training schemes, and the one I have used in the simulations presented in this thesis, is the so-called backpropagation that is applied to multi-layer perceptrons (MLPs) [28], [29].

The feedforward calculations are given by equation 3.4 and refers to Figure 3.3. The backpropagation algorithm used in my work is shown in Equation 3.7, and refers also to Figure 3.3.

$$y_l^{[out]} = y_l^{[3]} = f_l^{[3]}(u_l^{[3]}) = f_l^{[3]}(\sum_{k=1}^{n_2} a_k^{[2]} w_{kl}^{[3]}) = f_l^{[3]}(\sum_{k=1}^{n_2} f_k^{[2]}(u_k^{[2]}) w_{kl}^{[3]}) =$$

$$= f_l^{[3]}(\sum_{k=1}^{n_2} f_k^{[2]}(\sum_{j=1}^{n_1} f_j^{[1]}(u_j^{[1]}) w_{jk}^{[2]}) w_{kl}^{[3]}) = f_l^{[3]}(\sum_{k=1}^{n_2} f_k^{[2]}(\sum_{j=1}^{n_1} f_j^{[1]}(\sum_{i=1}^{N} x_i w_{ij}^{[1]}) w_{jk}^{[2]}) w_{kl}^{[3]}) \quad \text{(eq. 3.4)}$$

The procedure is based on the well known gradient descent method that is applied in the classic optimization procedures, on either an error $E_p$ found on a pattern by pattern (set of music features obtained from the analysis of the recorded scores) basis (online training) or on a total batch error $E$ (Sum Square Error, SSE) that is found for all the errors. The two errors are defined as shown in equations 3.5 and 3.6.

$$E_p = \frac{1}{2}\sum_{j=1}^{N_o}(d_{jp} - y_{jp,out})^2 = \frac{1}{2}\sum_{j=1}^{N_o} e_{jp}^2 \quad \text{(eq. 3.5)}$$

$$E = \frac{1}{2}\sum_{p}^{P} E_p = \frac{1}{2}\sum_{p}^{P}\sum_{j=1}^{N_o}(d_{jp} - y_{jp,out})^2 = \frac{1}{2}\sum_{p}^{P}\sum_{j=1}^{N_o} e_{jp}^2 \quad \text{(eq. 3.6)}$$

For a three layer MLP using the backpropagation algorithm, the weight updating is given by the following equations.

$$\Delta w_{ij}^{[3]} = -\eta \frac{\partial E_p}{\partial w_{ij}^{[3]}} = \eta \delta_j^{[3]} a_i^{[2]} \quad \text{(eq. 3.7a)}$$

$$\Delta w_{ij}^{[2]} = -\eta \frac{\partial E_p}{\partial w_{ij}^{[2]}} = \eta \delta_j^{[2]} a_i^{[1]} \quad \text{(eq. 3.7b)}$$

$$\Delta w_{ij}^{[1]} = \eta \delta_j^{[1]} x_i \quad \text{(eq. 3.7c)}$$

where $$\delta_j^{[2]} = \frac{\partial f_j^{[2]}}{\partial u_j^{[2]}} \sum_{i=1}^{n_3} \delta_i^{[3]} w_{ij}^{[3]}$$ (eq. 3.7d)

$$\delta_j^{[1]} = \frac{\partial f_j^{[1]}}{\partial u_j^{[1]}} \sum_{i=1}^{n_2} \delta_i^{[2]} w_{ij}^{[2]}$$ (eq. 3.7e)

More generally, the synaptic weight updating is done by equation …

$$w_{ij}^{[L]}[\kappa+1] = w_{ij}^{[L]}[\kappa] + \Delta w_{ij}^{[L]}[\kappa]$$ (eq. 3.8a)

where,

$$\Delta w_{ij}^{[L]}[\kappa] = \eta \delta_j^{[L]} a_i^{[L-1]} + \mu \Delta w_{ij}^{[L]}[\kappa-1]$$ (eq. 3.8b)

In equations 3.7 and 3.8, $\eta$ is the learning coefficient which controls the speed of learning. Normally should be high enough to attain fast convergence but at the same time not to make the system unstable. In eq. 3.8 $\mu$ is the so-called momentum coefficient that helps the network to avoid local minima in the error function.

## Support vector machines (SVM)

Support vector machines (SVM) were introduced in COLT-92 Conference on Learning Theory by Boser, Guyon and Vapnik. It originates in the statistical learning theory that received important impetus during the 60s [30], [31]. Ever since, there are a numerous of successful applications in many fields (bioinformatics, text recognition, image recognition, . . . ). It requires few examples for training, and is insensitive to the number of dimensions.

Essentially, SVM learn classification or regression mappings $\mathbf{X} \rightarrow \mathbf{Y}$, where $\boldsymbol{x} \in \mathbf{X}$ is some object and $\boldsymbol{y} \in \mathbf{Y}$ is a class label. In the general application area of pattern recognition they have been highly successful. For example, in a two-class classification problem, one way of representing the task is: for given $\boldsymbol{x} \in \boldsymbol{R}^n$ determine $\boldsymbol{y} \in \{+1, -1\}$. That is, just like all classification ML techniques, in a two-class learning task, the aim of a SVM is to find the best classification function that distinguishes between members of two classes in the training data.

In a similar manner to ANN and other ML tools, the training set is a set of $(x_1, y_1)$, … , $(x_m, y_m)$.

For the class separation, a hypercurve may be used. However, for a simple description a linearly separable dataset is considered. Then a linear classification function corresponds to a separating hyperplane $\boldsymbol{y}=f(\boldsymbol{x}, \boldsymbol{w})=\boldsymbol{w}\cdot\boldsymbol{x}+\boldsymbol{b}$, where $\boldsymbol{w}$ is a set of appropriate parameters, that splits the two classes, and thus separating them. There are many linear hyperplanes though. The SVM approach simply guarantees that the best such function is found by maximizing the margin between the two classes (Fig. 3.4).

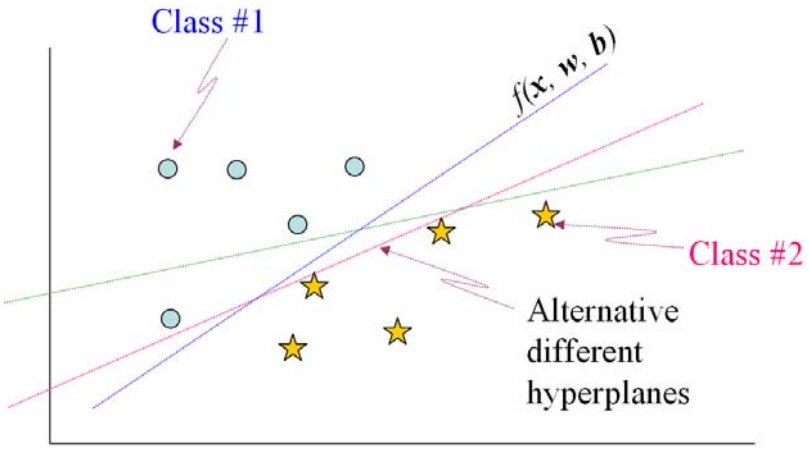


Figure 3.4: Alternative hyperplanes in a 2-class classification

The margin is defined as the amount of separation between the two classes. Thus the objective is to maximize this margin, through the use of appropriate optimization tools. The training however of SVMs is laborious when the number of training points is large. A number of methods however, for fast SVM training have been proposed. Thus the complexity issue is very important.

Based on the above simple explanations, the SVM may be generalized and formulated in the following algorithmic equations.

MaxMargin=minimize{Training Error + Complexity}=

$$=\arg\min\left\{\frac{1}{m}\right\}\sum_{i=1}^{m} d\left(f(\boldsymbol{x},\boldsymbol{w}),\ \boldsymbol{y}\right)+\text{Complexity term} \qquad \text{(eq. 3.13)}$$

Where $\boldsymbol{w}$ (weights) and $\boldsymbol{b}$ (biases) are appropriate adjustable parameters.

For the linear case, $\boldsymbol{y}=f(\boldsymbol{x},\ \boldsymbol{w})=\boldsymbol{w}\cdot\boldsymbol{x}+\boldsymbol{b}$, and the above reduces to:

$$\text{MaxMargin}=\arg\min\left\{\frac{1}{m}\right\}\sum_{i=1}^{m} d\left(\boldsymbol{w}\cdot\boldsymbol{x}+b,\ \boldsymbol{y}\right)+\|\boldsymbol{w}\|^2 \qquad \text{(eq. 3.14)}$$

subject to $\min_i=\left|\boldsymbol{w}\cdot\boldsymbol{x}\right|=1$

In the case where the map is not linearly separable, a new form is used as follows.

$$\underset{f,\ \xi_i}{\operatorname{argmin}}\left\{C\sum_{i=1}^{m}\xi_i+\|\boldsymbol{w}\|^2\right\} \quad \text{where } y_i(\boldsymbol{w}\cdot\boldsymbol{x}+\boldsymbol{b})\geq 1-\xi_i, \quad \text{for all } \xi_i\geq 0$$

The variables $\xi_i$ are called slack variables and they measure the error made at point $(\boldsymbol{x}_i,y_i)$.

There are many variations of the above that handle more complex and highly nonlinear problems.

## 3.4 Settings used in the various machine learning algorithms:

For each of the desired models, different algorithms were investigated in order to obtain different paradigms from each algorithm and eventually choose the algorithm that gives the best accuracy in the predictions. All the models were build using the Weka environment. For the regression, the algorithms used were:
a) Support vector machines
b) K-nearest neighbours
c) Artificial neural networks
d) M5 Rules of regression trees

For classification the algorithms that were used were:
a) Support vector machines
b) K-nearest neighbors
c) Artificial neural networks
d) J48.

In this section, the various settings of each algorithm that have been used are briefly presented.

Support vector machines settings.

Two models were build using the support vector machines algorithm. The first model was using the first degree polynomial kernel while the second model was using the second degree polynomial kernel.

The filter type used was the "normalized training data" and the epsilon was 1.0E-12. The epsilon parameter was 0.0010 and the tolerance was 0.0010.

K-nearest neighbours settings.

The algorithm was trained using only one-nearest neighbour. No distance weighting was used and the distance function employed was the Euclidean Distance.

Artificial neural network settings

One hidden layer MLP structure was used for the training. The learning rate was 0.3 and the momentum was 0.2. The training time was 500 epochs. The validation set size was 0 and the validation threshold was 20.

M5 Rules settings.

The minimum number of instances that was used was 4.
No debugging, unpruningn and unsmoothing was used.

J48 settings.

The confidence factor was 0.25, the minimum number of objects was 2 while the number of folds was 3.
No binary splits, debugging, reduced error pruning was used.

# 4. Audio feature extraction

## 4.1 Data

Two sets of data were collected and used in the investigations reported in this thesis. Both sets were recorded in the well-equipped studio that is located in the campus of the University of Pompeu Fabra.

The first data set consists of monophonic violin performances of four pieces, each one performed with four different emotions. The pieces were:

(a) “La Comparsita” written by Gerardo Matos Rodríguez in 1917 consisted by 69 notes,
(b) “Largo Invierno” composed by Antonio Vivaldi consisted by 76 notes,
(c) “Por una Cabeza” composed by Carlos Gardel and Alfredo Le Pera in 1935 consisted by 92 notes, and
(d) “La Primavera” composed by Antonio Vivaldi consisted by 98 notes.

The emotions expressed and recorded in “La Comparsita” were “angry”, “happy”, “sad” and “fear”, while for the other three pieces the emotions were “angry”, “happy”, “sad” and “sweet”.

The second data set consisted of monophonic tenor saxophone performances of three jazz pieces each one performed with four different emotions. The pieces were:

(a) “Boblicity” recorded by Miles Davies in 1949 consisted by 173 notes,
(b) “How deep is the ocean” written by Irving Berlin in 1932 consisted by 93 notes, and
(c) “Lullaby of birdland” composed by George Shearing in 1952 consisted by 152 notes.

The emotions that the pieces were recorded were “angry”, “happy”, “sad” and “fear”.

## 4.2 Note segmentation

In order to obtain a symbolic representation of the recorded performances, signal processing techniques were applied to the audio recordings. The procedure for obtaining such symbolic description is described below. First, the audio signal is divided into analysis frames, and a set of low-level descriptors are computed for each analysis frame. Then, note segmentation is performed by using low-level descriptor values. These descriptors are the energy and the fundamental frequency. Both results are merged to find the note boundaries. A schematic diagram of this process is shown in Figure 4.2a.

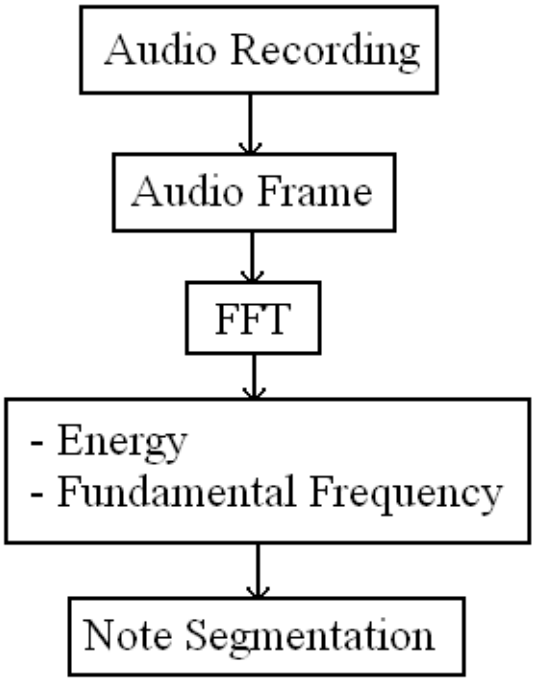


Figure 4.2a. Low level descriptors computation and note Segmentation

More specifically, the energy values were stored in a vector and the time derivative was computed in order to identify the peaks in the vector, which occur when there are fast changes in the signal, as it is the case when there is an attack of a note. After that, a simple peak detection algorithm has been applied to that vector, using a given threshold. These peaks will be later used in order to decide if the position of the peak is a starting note or not. Fundamental frequency values were computed for each frame using the Yin algorithm [32], and once again the derivative has been re-computed in order to extract differences in the pitch which correlate with a changing note.

Finally, a routine that merges neighbouring onsets that are too close has been implemented, by erasing multiple peaks that belong to the same 'lobe'. This algorithm iteratively scans the curve for peaks from start to finish and vice versa, and erases them if there is a higher peak between two frames of the Yin analysis. Finally, all onsets that were detected for areas where the RMS energy was lower than a given auditory threshold, were discarded, in order to avoid false onsets.

Figure 4.2b shows a typical fundamental frequency vector, Figure 4.2c the energy variation, and Figure 4.2d the frequency variation. Figure 4.2e shows the onsets based on frequency, Figure 4.2f the onsets based on the energy and Figure 4.2g the combined onsets.

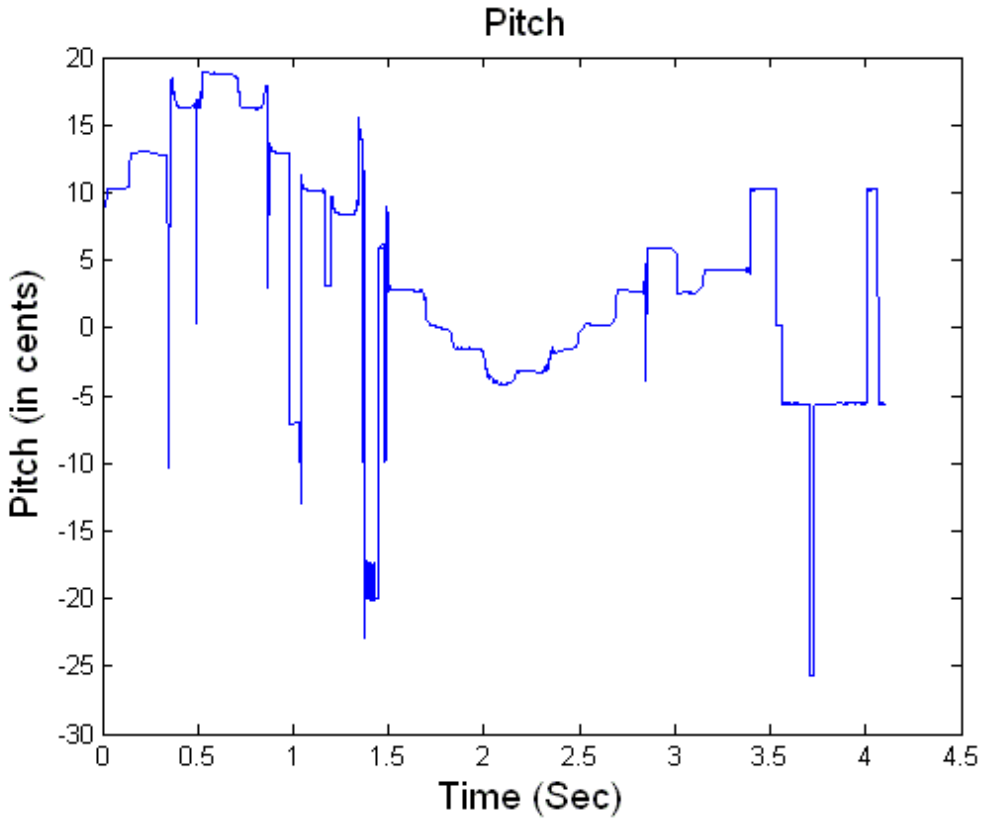


Figure 4.2b. Typical fundamental frequency vector

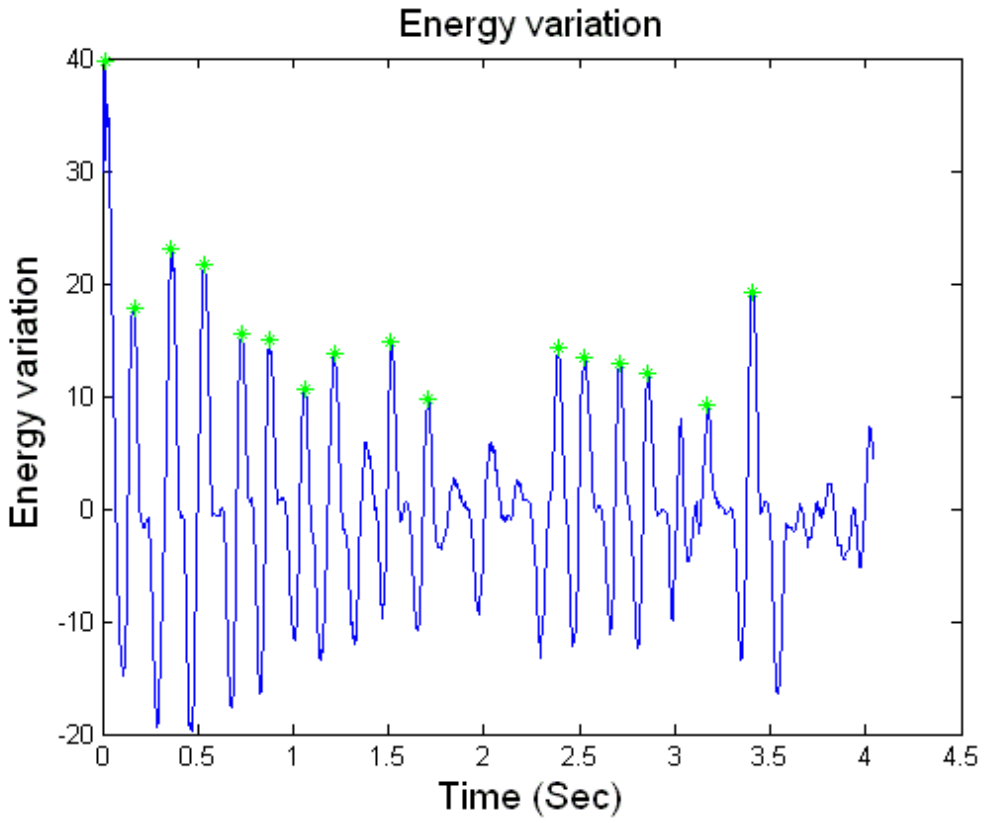


Figure 4.2c: Energy variation

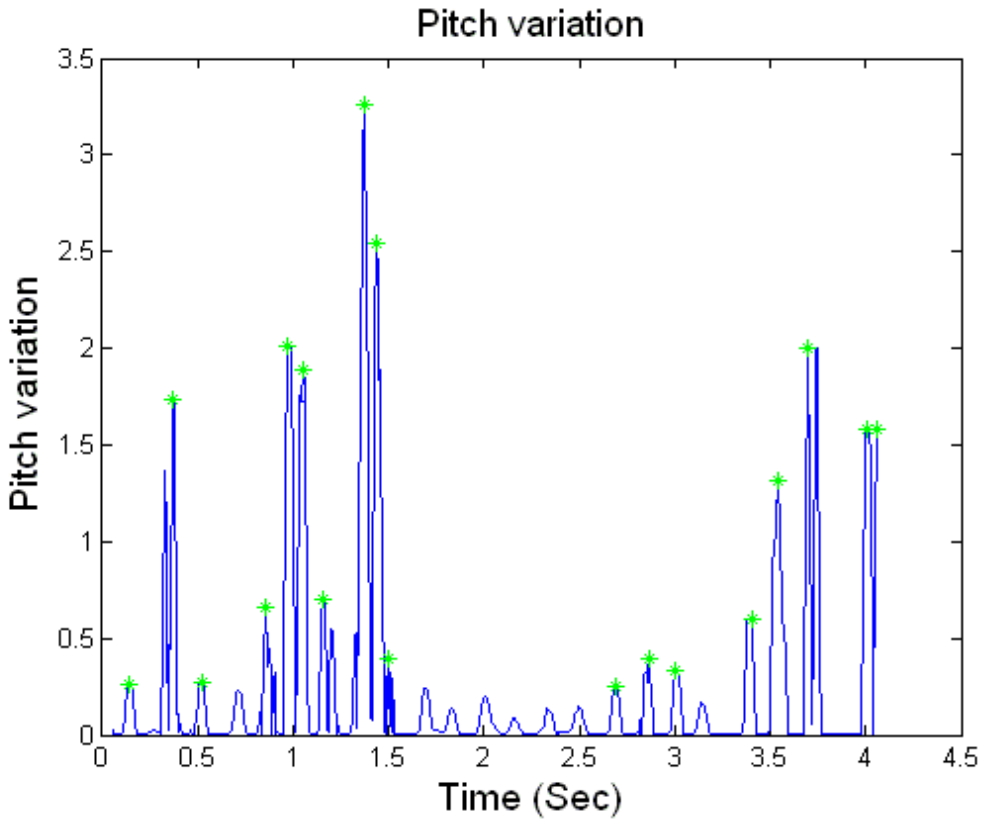


Figure 4.2d. Pitch variation

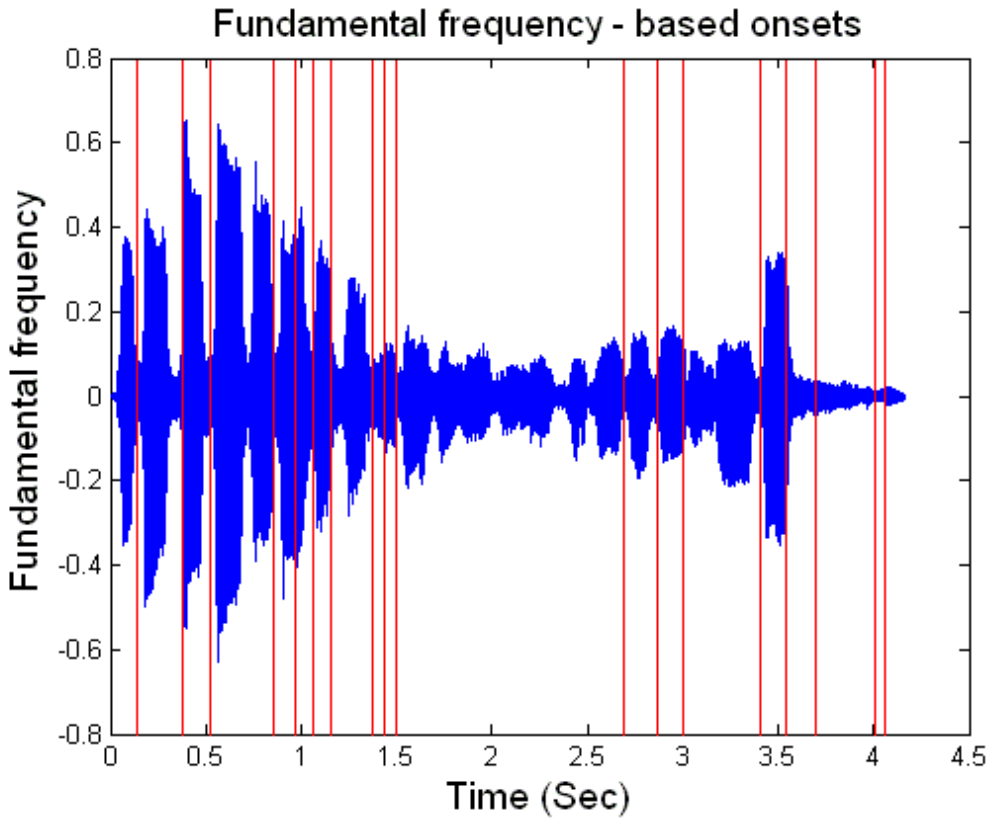


Figure 4.2e. Onsets based on frequency

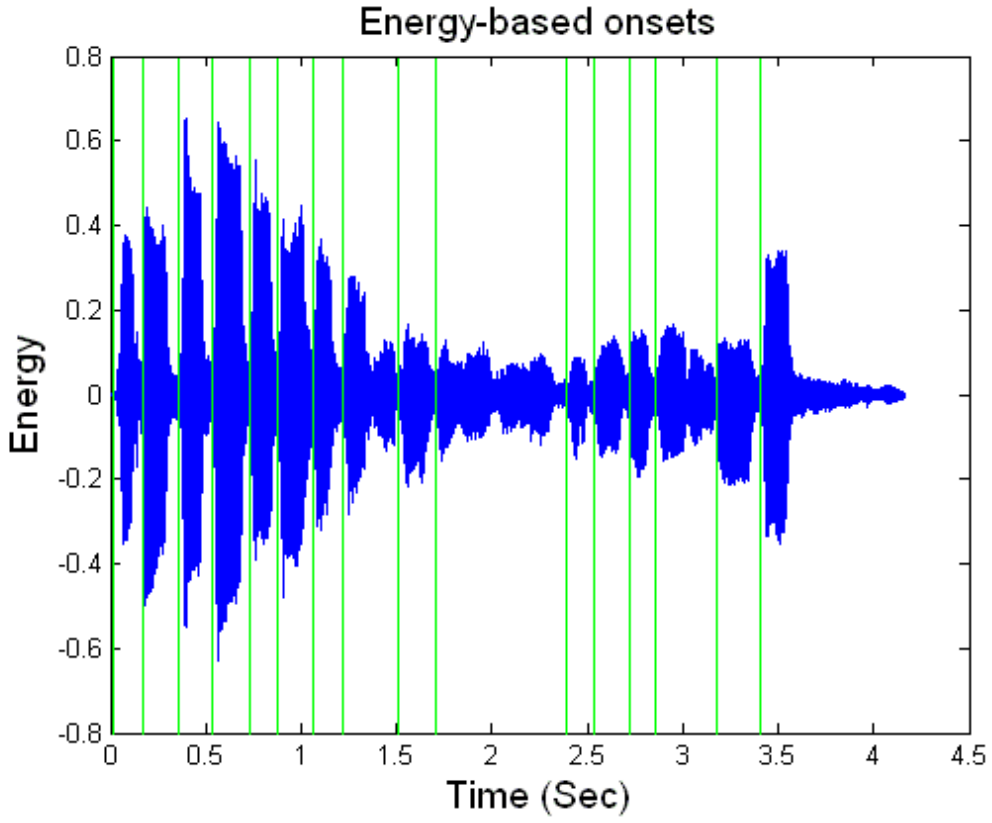


Figure 4.2f. Onsets based on energy

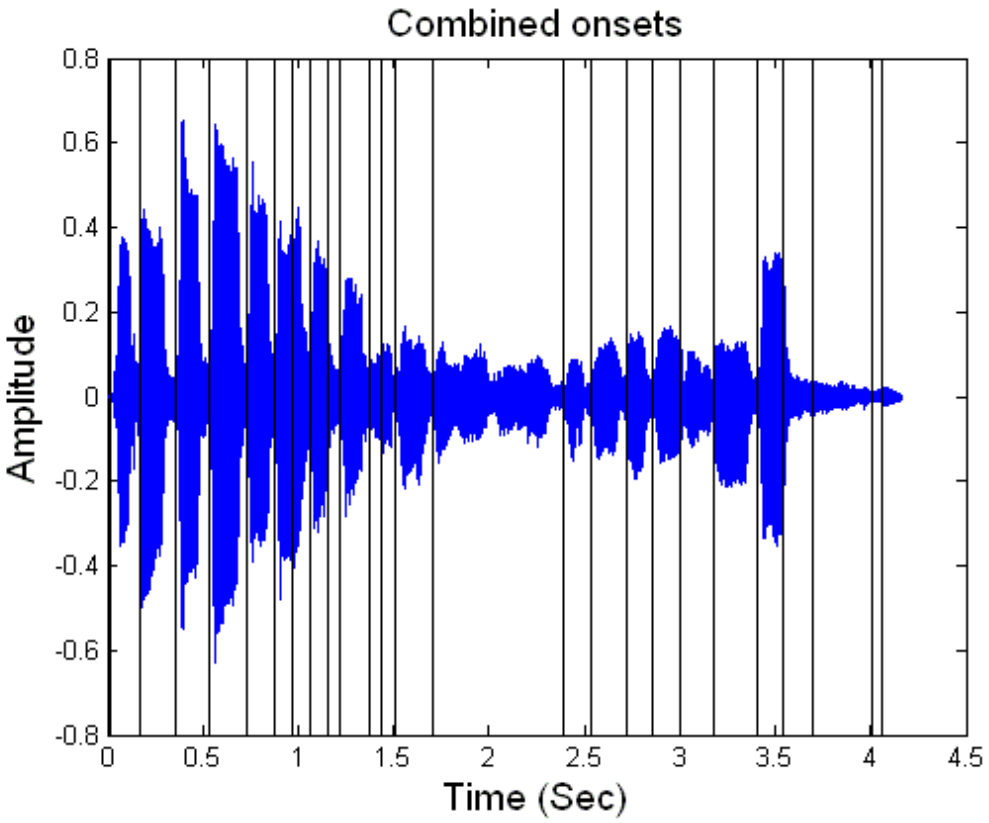


Figure 4.2g. Combined onsets

## 4.3 Features

Once the note boundaries are known, a set of note descriptors have been computed and these descriptors have been used as input features for the algorithms. Information about intrinsic properties of the note includes the note duration and the note metrical position, while information about its context includes duration of previous and following notes, extension and direction of the intervals between the note and the previous and the following notes, and the note Narmour group(s) [25]. The Narmour's Implication/Realization model is a theory of perception and cognition of melodies. The theory states that a melodic musical line continuously causes listeners to generate expectations of how the melody should continue. Any two consecutively perceived notes constitute a melodic interval and if this interval is not conceived as complete, it is an *implicative interval*, i.e. an interval that implies a subsequent interval with certain characteristics. Figure 4.3a shows prototypical Narmour structures. A note in a melody often belongs to more than one structure, i.e. a description of a melody as a sequence of Narmour structures consists of a list of overlapping structures. Each melody is parsed in the training data in order to automatically generate an implication/realization analysis. All these features will be later used as attributes in order to build the models using machine learning algorithms. Results of the learning process will be analyzed and the set of features involved will be refined accordingly.

For synthesis purposes I am concerned to build models and predict values for note duration and note energy expressive transformations and also to predict the bow direction for the violin performances. For the saxophone performances I am concerned to build models and predict values for note duration and note energy. Each note in the training data is annotated with its corresponding deviation and bowing direction and a number of attributes representing both properties of the note itself and some aspects of the local context in which the note appears. Bow direction is computed by finding the derivatives of the bow position. By computing the two derivatives of the bow position we have the velocity and the acceleration. The bow position is computed as the Euclidian distance (in cm) between Pi point of contact of the bow and the string and the frog part of the bow which is in the beginning of the bow, where the hair starts. The range of values goes from close to zero at the frog to around 65 cm at the tip (depending on the length of the bow). During string changes, the point of contact bow-string changes suddenly, producing discontinuities in the values of the bow position, which in turn causes erroneous values of its derivatives (bow velocity and bow acceleration). In this way is computed the bow direction.

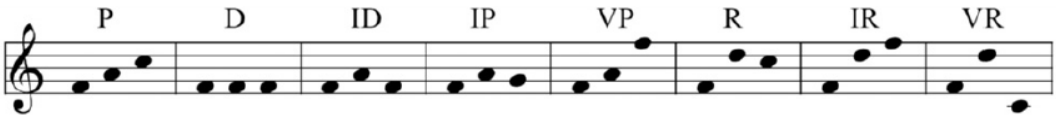


Figure 4.3a Prototypical Narmour structures.

# 5. Results and discussion

The data were analyzed in two different ways. First, the transformations of the performances, in duration and in energy, were plotted for better visualization. In the same figure, the model transformations were also plotted. In this way, it was easier to see how close the prediction values were to the real ones. The main descriptors used were the mean energy and the duration ratio of the notes between the score and the performances. Also, temporal and frequency descriptors, for instance attack level, attack slope and spectral centroid were used. In the cases studied, even though the descriptors were used in the training, they could not be used in the synthesis because the synthesizer could not take them as inputs. The synthesizer uses only the duration and the pitch. For the case of the violin synthesis, the bow direction is also taken into consideration. Furthermore, the pitch was not predicted, but it was sent to the synthesizer as it is.

As mentioned before, the values from the predictions had to be as close as possible to the real values in order to have realistic representation of the descriptors analyzed. For instance, the duration of the prediction has to be very close to the real duration. In that case, and after the synthesis, the sound has to be perceptually the same or very similar to the original sound.

## 5.1 Cross-validation results

The correlation coefficients for the duration and energy models and the percentage of correctly classified instances for the bow direction model for the song “Comparsita” are shown in Tables 5a, 5b and 5c. The values shown were obtained by performing a ten-fold cross-validation on the data. At each fold, the notes similar to the ones selected in the test set were removed, that is, the notes of repetitions of fragments in the test set. The songs were separated into four phrases. Each phrase has musical meaning and is consisted from approximately 15 – 20 notes each. Five different models were build using different combinations by removing one phrase from the overall data. The phrase that was removed was later used as the test set.

As shown in the Tables 5a and 5b, the correlation coefficients for the duration ratio and for the energy for the “sad” emotion is low and this is probably because the recordings were done with a metronome and in that case is always difficult for the musician to express the sad emotion when he/she is restricted by the metronome. In that case it affects the performance, and therefore the duration and the energy which is linearly related to the dynamics of the notes are not the desirable ones.

Table 5.1a. Ten-fold cross-validation correlation coefficients for the **duration ratio** for the emotions angry, fear, happy and sad for phrase one, for the **"Comparsita"** song.

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.8159** | 0.6836 | 0.3937 | 0.6882 | 0.6159 |
| Fear | 0.6687 | 0.7328 | 0.487 | 0.7135 | **0.7775** |
| Happy | 0.6697 | **0.7269** | 0.4627 | 0.6244 | 0.6971 |
| Sad | 0.2089 | **0.3821** | 0.3102 | 0.0777 | 0.1815 |

Table 5.1b. Ten-fold cross-validation correlation coefficients for the **energy** for emotions angry, fear, happy and sad for phrase two for the song **"Comparsita"**.

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.3266 | 0.5348 | 0.1525 | **0.6307** | -0.2955 |
| Fear | 0.2146 | **0.5509** | 0.063 | 0.3929 | 0.4043 |
| Happy | 0.387 | **0.5464** | 0.2237 | 0.4243 | 0.3406 |
| Sad | 0.2889 | **0.3859** | 0.1586 | 0.2305 | 0.2999 |

Table 5.1c. Ten-fold cross-validation correctly classified instances percentage for the **bow direction** for emotions angry, fear, happy and sad for phrase four for the song **"Comparsita"**.

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 89.33 | 90.67 | 87.33 | **91** | 87.33 |
| Fear | **96.33** | 90.67 | 94.67 | 94.67 | 94.67 |
| Happy | **88.33** | 82.67 | 86.33 | 84.67 | 84.67 |
| Sad | 83.33 | **85.33** | 81.33 | 81.33 | 81.33 |

The predictions of the duration ratio for the song "Largo Invierno", for the emotions of "angry" and "happy" and for phrase one was less than 0.5, while the predictions for the emotions "sad" and "sweet" were above 0.5. This means that the algorithms could not learn the transformations of the performer for the two emotions, for this part of the song. The predictions of the energy, for the emotions "angry" and "happy" were above 0.5, while for the emotions "sad" and "sweet" were below 0.5, which was the case for most of the times. The predictions of the bow direction were above 50% for all the emotions. This can be seen in the Tables B.as and B.at that are shown in Appendix B.

The predictions of the duration ratio and the energy for the song "Por una cabeza" were below 0.5 for all the phrases and for all the emotions. This means that some of the features had no consistency, or they did not give any useful information to the algorithms in order to learn correctly. This can be seen in the Tables B.be – B.bs in the Appendix B.

The predictions of the duration ratio and the energy of the song "Primavera" and "Boblicity" were most of the times above 0.5 for all the phrases and for all the

emotions. That means that the algorithms were able to learn and predict the transformations of the performer. This can be seen in the tables B.bu – B.cr in the Appendix B.

The predictions of the duration ratio and the energy of the song “How deep is the ocean” and “Lullaby of birdland” are most of the times close to 0.5. For instance the best correlation coefficient of the duration ratio of the song “How deep is the ocean” for the phrase two and for the “angry” emotion is 0.72 while the best correlation coefficient for the same emotion but for phrase three is 0.26. This can be seen in the Tables 1cw and 1cy in the Appendix B. Similar results have been observed for all the phrases of that song. This means that some phrases were difficult for the algorithms to learn the transformations of the performer. The correlation coefficients for the songs “How deep is the ocean” and “Lullaby of birdland” can be seen in the Appendix B in the Tables B.cu – B.dj.

## 5.2 Performance-predicted comparison

Figures 5.2a to 5.2d show the note-by-note duration ratio predicted by the best model and compared with the actual duration ratio in the recording for angry, happy, fear and sad emotions for the song "Comparsita". As illustrated by Figure 5.2a – 5.2d, the induced models seem to accurately capture the musician's expressive transformations.

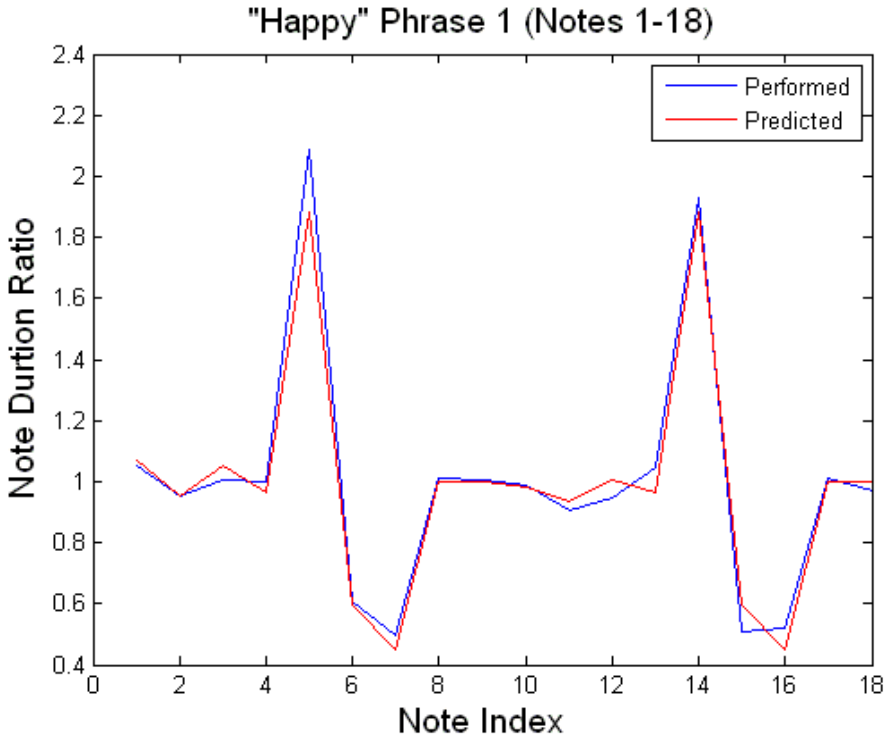


Figure 5.2a. Comparison between the **duration ratio** of the model's transformation predictions and the actual transformations performed by the musician for the "happy" mood for the song **"Comparsita"**. The test set was removed from the training set.

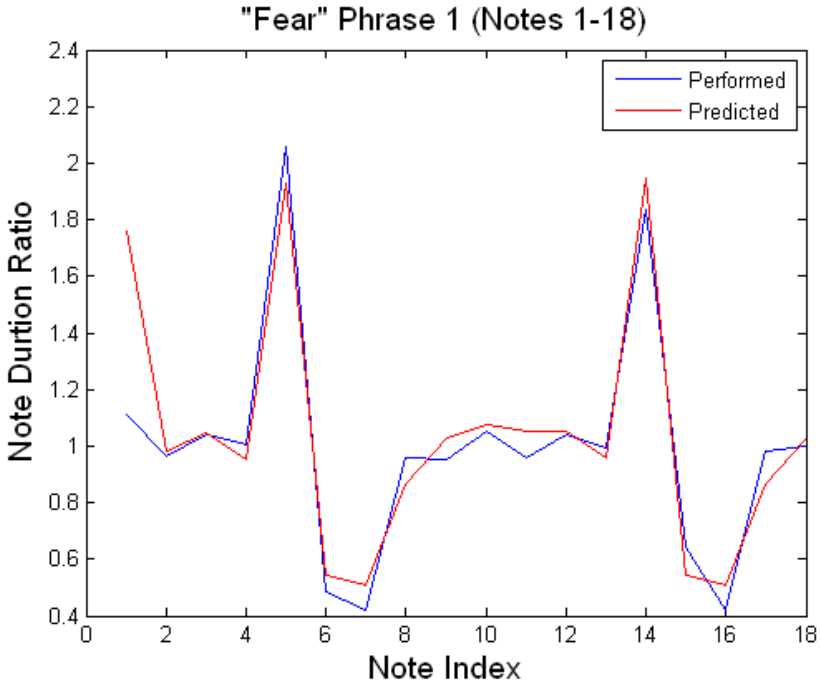


Figure 5.2b. Comparison between the **duration ratio** of the model's transformation predictions and the actual transformations performed by the musician for the "fear" mood for the song **"Comparsita"**. The test set was removed from the training set.

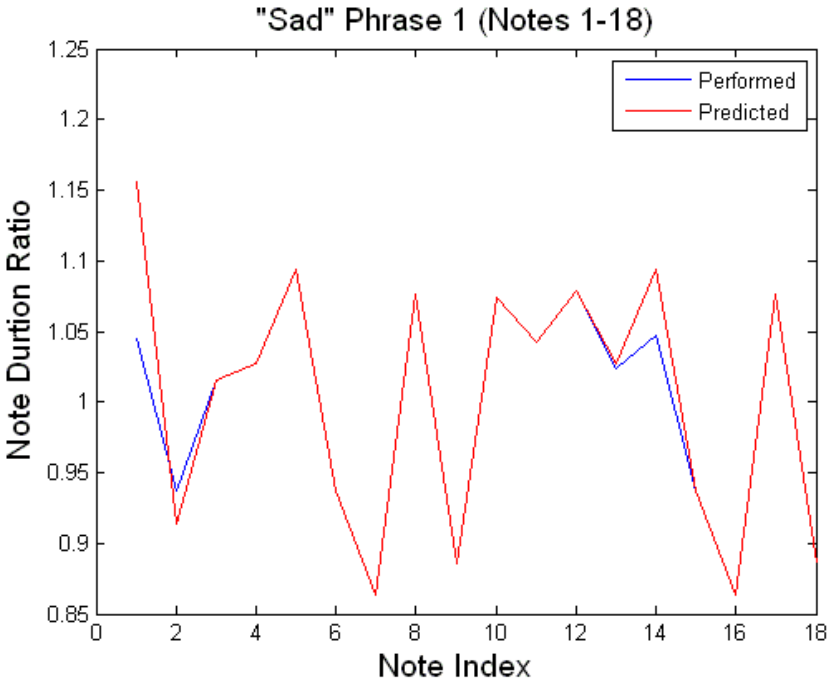


Figure 5.2c. Comparison between the **duration ratio** of the model's transformation predictions and the actual transformations performed by the musician for the "sad" mood for the song "**Comparsita"**. The test set was removed from the training set.

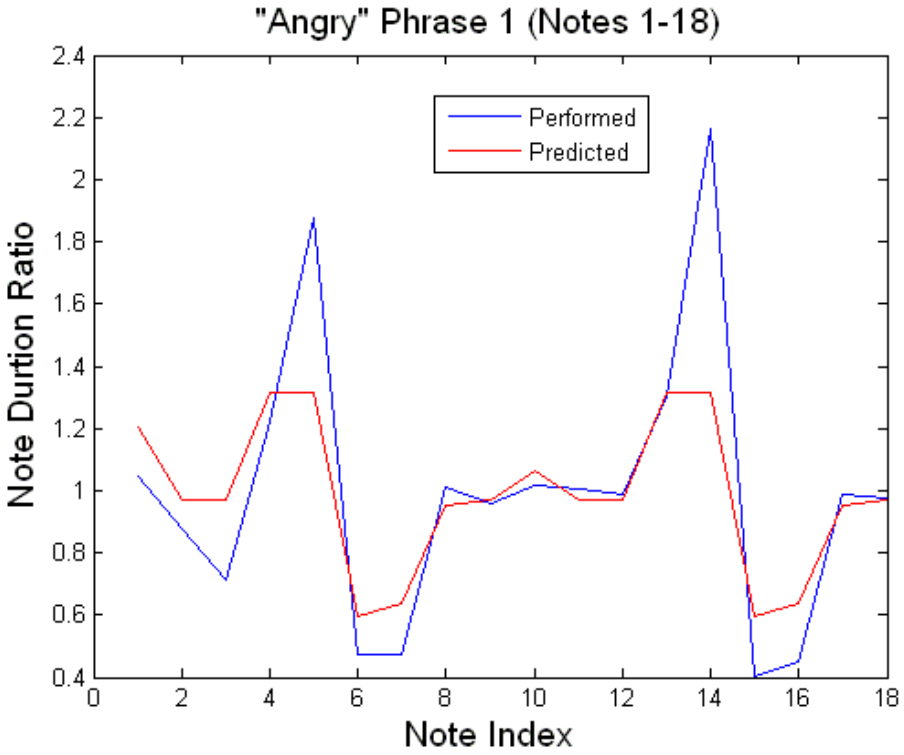


Figure 5.2d. Comparison between the **duration ratio** of the model's transformation predictions and the actual transformations performed by the musician for the "angry" mood for the song **"Comparsita"**. The test set was removed from the training set.

## 5.3 Perceptual evaluation

The second method of evaluation was to ask people to hear pairs of songs and judge which ones are from human performance or computer generated. Seven students of the Master in Sound and Music Computing were asked. These were between twenty two and thirty three years old, and no one had hearing problems. They were asked to mark which music style they liked out of Rock, Classical, Jazz, Hip-hop/Rap, Electroacoustic, Folk, Heavy metal and Techno.

The questionnaire (shown in Appendix A) was consisted of five pairs of melodies arranged in two sections. In the first section, for each melody pair, one melody was synthesized by information coming from the human performer (duration of the notes, pitch and bow direction) and the other melody that was synthesized from information coming from the score. In this case, the goal was to be easy for people to understand and identify which one is from a human performer and which one was synthesized from the score. In this sense, the information used to synthesize the melodies is informative enough to discriminate between the nominal score and the expressive performance.

Table 5.3d shows that most of the pairs were correctly answered except of the third pair. The overall percentage of the correct answers is 62.86%.

In the second section, in each pair, one of the melodies was synthesized from information coming from the human performance and the other melody was

synthesized from the information coming from the prediction of the model. In this case, the goal is to be difficult for people to understand which piece is from human performance and which one generated by a computer, based on the information from the prediction. The highest is the percentage in the wrong answers, the better it is to achieve the goal. Table 5.3e shows that most of the pairs were wrongly answered except the fourth pair for which the percentage of wrong answers was 42.86%. The overall percentage in the wrong answers is 71.43%.

Table 5.3d. Percentage of correct answers for the pair of human performance and synthesized score. The subjects were asked to mark the human performance.

| **Section 1** | correct | wrong | % correct |
|---|---|---|---|
| Pair 1 | 6 | 1 | 85.72% |
| Pair 2 | 4 | 3 | 57.14% |
| Pair 3 | 2 | 5 | 28.57% |
| Pair 4 | 5 | 2 | 71.43% |
| Pair 5 | 5 | 2 | 71.43% |
| **Overall** | 22 | 13 | **62.86%** |

Table 5.3e. Percentage on the correct answers for the pair of human performance and the computer generated. The subjects were asked to mark the computer generated.

| **Section 2** | correct | wrong | % Wrong |
|---|---|---|---|
| Pair 1 | 2 | 5 | 71.43% |
| Pair 2 | 2 | 5 | 71.43% |
| Pair 3 | 1 | 6 | 85.72% |
| Pair 4 | 4 | 3 | 42.86% |
| Pair 5 | 1 | 6 | 85.72% |
| **Overall** | 10 | 25 | **71.43** % |

# 6. Conclusions and future work

## Conclusions:

In this master thesis I presented a machine learning approach to modelling musical emotional expression in violin and saxophone performances. I investigated how professional musicians encode emotions such as happiness, sadness, anger, fear and sweetness in their performances. My objective has been to find a computational model that predicts how a particular note in a particular context should be played (e.g., longer or shorter than its nominal duration) in order to express various sentiments. In order to induce expressive performance knowledge, I have extracted a set of acoustic features from the recordings resulting in a symbolic representation of the performed pieces. I then applied both classification and regression methods to the symbolic data and information about the context in which the data appears. The predictions of the models were high enough to obtain good results for the synthesis. In only one song the predictions of all the models and for all the emotions were low (for the song "Por una cabeza"). Most of the synthesized songs were perceptually difficult to understand that is synthesized by the computer.

## Future work:

For future work, an extension of this work in order to pursue further classification on what is the emotion of a set of songs by using suitable models of different emotions will be attempted. This can be done through the use of statistical tools as well as modern machine learning classification tools (e.g. support vector machines, artificial neural networks, k-means, radial basis functions, decision trees, evolutionary algorithms, etc), and through the application of appropriate comparison measures. Thus, the aim will be to find the closest distance and conclude what is the closest mood for each new song. To make it clearer, in Figure 6a an example is shown, where a new song is characterized to be closer to sad and fear than to happy and angry.

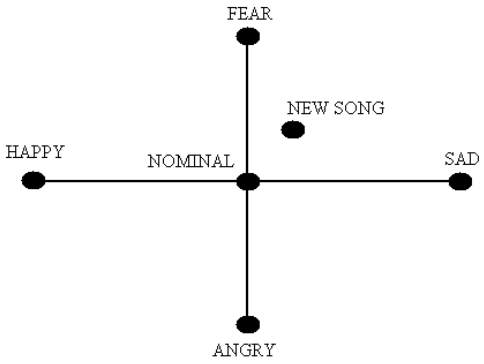


Figure 6a. Automatic emotion classification of an unknown song.

Another interesting extension to explore is to model the interaction among musicians when they play together. It is known from musicology studies e.g. [33, 34] that musicians interact together in terms of emotions and this affects the individual, as well the overall performance. A musician, almost always performs a song in a different way when he/she performs a song on his/her own and in a different way

when he/she performs a piece when playing with other musicians. This is due to psychological issues as well as the real-time interaction between musicians. The stimuli comes automatically because when people perform music, the emotions that they express are not coming only by the sound they produce, but also by the gestures and the movements of their body. Often musicians complain after a concert that their performance was not good because a member of the group was not in a good mood and as a result, this member drifted the other members of the group to bad performances. On other occasions though, the opposite may happen.

Another interesting area to investigate is the style-based performer identification. In this context, expressive models of several different musicians can be built. Following that, style-based predicting systems that automatically identify a performer can be developed (out of a group of possible performer choices), by taking into account the expressive style of the performers. This can be very interesting for investigating the differences between the performances of different musicians, classify them and build models according to the way that each one performs particular songs. This can be extended to multi-performer style-based identification. This is the study of the way that different groups of musicians perform together and the use of the resulting information to automatically identify the group of musicians playing a piece.

# References

[1] A. Gabrielsson, "The performance of music," *In: Deutsch, D. (Ed.), The Psychology of Music, second ed. Academic Press*, 1999.

[2] G. Widmer, "Discovering Strong Principles of Expressive Music Performance with the PLCG Rule Learning Strategy;" *Proceedings of the 12th European Conference on Machine Learning (ECML'01),* Freiburg, Germany. Berlin: Springer Verlag. 2001.

[3] M.J. Dovey, "Analysis of Rachmaninoff's Piano Performances Using Inductive Logic Programming," *European Conference on Machine Learning*, Springer-Verlag. 1995.

[4] R. Ramirez, A. Hazan, E. Maestre, X. Serra, "A Genetic Rule-Based Model of Expressive Performance for Jazz Saxophone," *Computer Music Journal*, 2008.

[5] R. Ramirez, A. Perez, S. Kersten, D. Rizo, P. Román, J.M. Iñesta, "Modeling Violin Performances Using Inductive Logic Programming, Intelligent Data Analysis," 2010.

[6] Repp, B.H. "Diversity and Commonality in Music Performance: an Analysis of timing Microstructure in Schumann's 'Traumerei'," *Journal of the Acoustical Society of America 104,* p.p. 2546-2568, 1992.

[7] Todd, N. "The dynamics of Dynamics: a Model of Musical Expression," *Journal of the Acoustical Society of America,* Vol. 91, 1992.

[8] A. Friberg, R. Bresin, L. Fryden, J. Sundberg: "Musical Punctuation on the Microlevel: Automatic Identification and Performance of Small Melodic Units," *Journal of New Music Research,* 27(3): p.p. 217-292, 1998.

[9] A. Friberg, V. Colombo, L. Fryden, J. Sundberg, "Generating music performances with director musices," *Computer Music Journal,* p.p. 23-29, 2000.

[10] J. Sundberg, A. Friberg, R. Bressin, "Attempts to Reproduce a Pianist's Expressive Timing with Director Musices Performance Rules," *Journal of New Music Research,* p.p. 317-325, 2003.

[11] R. Bressin, G. Battel, "Articulation Strategies in Expressive Piano Performance Analysis of Legato, Staccato, and Repeated Notes in Performances of the Andante Movement of Mozart's Sonata in G Major," *Journal of New Music Research,* p.p. 211-224, 2000.

[12] R. Bressin, G. Widmer, "Production of staccato articulation in Mozart sonatas played on a grand piano.Preliminary results," *TMH-QPSR,* p.p. 001-006, 2000.

[13] M.L. Johnson, "An expert system for the articulation of Bach fugue melodies," *In: Baggi, D.L. (Ed.), Readings in Computer-Generated Music. IEEE Computer Society,* pp. 41–51, 1992.

[14] S. Canazza, G. De Poli, A. Roda, A. Vidolin, "Analysis and synthesis of expressive intention in a clarinet performance," *In: Proc. 1997 Internat. Comput. Music Conf.. International Computer Music Association, San Francisco,* pp. 113–120, 1997.

[15] R.D. Dannenberg, H. Pellerin, Derenyi, "A study of trumpet envelopes," *In: Proc. Internat. Comput. Music Conf. (ICMC), San Francisco,* 1998.

[16] R. Lopez de Mantaras, J.L. Arcos, "AI and music, from composition to expressive performance," AI Mag. 23 (3), 2002.

[17] R. Ramirez, A. Hazan, "A tool for generating and explaining expressive music performances of monophonic Jazz melodies," *Internat. J. Artif. Intell. Tools 15 (4),* p.p. 673–691*,* 2006.

[18] E. Van Baelen, L. De Raedt, "Analysis and prediction of piano performances using inductive logic programming," *In: Internat. Conf. in Inductive Logic Programming,* p.p. 55–71, 1996.

[19] A. Tobudic, G. Widmer, "Relational IBL in music with a new structural similarity measure," *In: Proc. Internat. Conf. on Inductive Logic Programming. Springer-Verlag,* 2003.

[20] G. Widmer, "Machine discoveries: A few simple, robust local expression principles," *Computer Music Journal*, 2002.

[21] C. Saunders, D. Hardoon, J. Shawe-Taylor, G. Widmer, "Using string kernels to identify famous performers from their playing style," *In: Proc. 15th Eur. Conf. on Machine Learning (ECML'2004), Pisa, Italy,* 2004.

[22] E. Stamatatos, G. Widmer, "Automatic identification of music performers with learning ensembles," Artif. Intell. 165 (1), p.p. 37–56, 2005.

[23] M. Grachten, G. Widmer, "Who is who in the end? Recognizing pianists by their final ritartandi," *10th International Society for Music Information Retrieval Conference (ISMIR),* 2009.

[24] R. Ramirez, E. Maestre, A. Pertusa, E. Gomez, X. Serra, "Performance-based Interpreter Identification in Saxophone Audio Recordings," *IEEE Transactions on Circuits and Systems for Video Technology,* p.p. 356-364, 2007.

[25] E. Narmour, "The analysis and cognition of basic melodic structures: The implication realization model," *University of Chicago Press,* 1990.

[26] R. Ramirez, A.Perez, S. Kersten, E. Maestre, "Performer identification in celtic violin recordings," *International Conference on Musing Information Retrieval,* 2008.

[27] S. Haykin, Neural Networks, *2nd Edition, Prentice Hall, 1999, ISBN 0 13 273350 1*

[28] P. Werbos "Beyond regression: New tools for prediction and analysis in the behavioral sciences," *Ph.D. Dissertation. Harvard University,* 1974.

[29] D. E. Rumelhart, G. E. Hinton and J. L. McClelland. In McClelland J. L., Rumelhart D. E. and the PDP Research Group (Eds.). "Parallel Distributed Processing: Explorations in the Microstructure of Cognition," *Vol. 1. Foundations. Cambridge, MA: MIT Press,* 1986.

[30] V. Vapnik, S. Golowich, and A. Smola. "Support vector method for function approximation, regression estimation, and signal processing." In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9, pages 281– 287, Cambridge, MA, 1997. MIT Press.*

[31] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", *Cambridge University Press,* 2000.

[32] A. de Cheveigne, H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Acoustical Society of America,*2002.

[33] D. Huron "Music and the Psychology of Expectation" *A Bradford Book The MIT Press Cambridge, Massachusetts London, England.*

[34] P. Juslin "Five Facets of Musical Expression: A Psychologist's Perspective on Music Performance" *Psychology of Music,* 2003.

# Appendix A

Questionnaire for the evaluation:

## Please answer the following questions:

Age: Male/Female: Number of years of music instruction:

Do you have a hearing problem?

No, Yes, specify

I like the following music style:

Classical, Rock, Jazz, Electroacoustic, Hip-hop/rap, Other, specify

Please listen to the following pairs of melodies. One of them is a human performance and the other is the synthesized score. Please mark which one is the human performance.

Pair 1:

Melody 1 Melody 2 I don't know

Pair 2:

Melody 1 Melody 2 I don't know

Pair 3:

Melody 1 Melody 2 I don't know

Pair 4:

Melody 1 Melody 2 I don't know

Pair 5:

Melody 1 Melody 2 I don't know

Please listen to the following pairs of melodies. One of them is a human performance and the other is computer generated. Please mark which one is the computer generated.

Pair 1:

Melody 1 Melody 2 I don't know

Pair 2:

Melody 1 Melody 2 I don't know

Pair 3:

Melody 1 Melody 2 I don't know

Pair 4:

Melody 1 Melody 2 I don't know

Pair 5:

Melody 1 Melody 2 I don't know

# Appendix B

Correlation coefficients for the duration and energy models and the correctly classified instances percentage for the bow direction of the song "Comparsita":

Table B.aa. Correlation coefficients of the **duration ratio** for the emotions "angry", "fear", "happy" and "sad**" when train and predict with the same data** for the song **"Comparsita".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.8407** | 0.7177 | 0.542 | 0.7284 | 0.7458 |
| Fear | **0.8158** | 0.7794 | 0.5485 | 0.7538 | 0.7927 |
| Happy | 0.705 | 0.7799 | 0.5207 | 0.7347 | **0.816** |
| Sad | 0.3369 | **0.5235** | 0.3846 | 0.2337 | 0.2738 |

Table B.ab. Correlation coefficients of the **energy** and for the emotions "angry", "fear", "happy" and "sad" **when train and predict with the same data** for the song **"Comparsita".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.4066 | **0.5603** | 0.2321 | 0.366 | 0.3633 |
| Fear | 0.4311 | **0.5611** | 0.0886 | 0.3133 | 0.4315 |
| Happy | 0.1312 | **0.6438** | 0.1075 | 0.144 | 0.1353 |
| Sad | 0.1958 | **0.442** | 0.1809 | 0.0673 | 0.2422 |

Table B.ac. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" **when train and predict with the same data** for the song **"Comparsita".**

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 80.71**%** | 80.95**%** | **82.62%** | 79.76**%** | **82.62%** |
| Fear | 75**%** | 76.67**%** | **79.29%** | 75**%** | 76.43**%** |
| Happy | **86.9%** | 75.48**%** | 82.86**%** | 78.33**%** | 82.86**%** |
| Sad | **88.57%** | 84.29**%** | 80**%** | 81.43**%** | 81.43**%** |

Table B.ad. Correlation coefficients of the **duration ratio** and for the emotions "angry", "fear", "happy" and "sad" and for **phrase one** for the song **"Comparsita".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.8159** | 0.6836 | 0.3937 | 0.6882 | 0.6159 |
| Fear | 0.6687 | 0.7328 | 0.487 | 0.7135 | **0.7775** |
| Happy | 0.6697 | **0.7269** | 0.4627 | 0.6244 | 0.6971 |
| Sad | 0.2089 | **0.3821** | 0.3102 | 0.0777 | 0.1815 |

Table B.ae. Correlation coefficients of the **energy** and for the emotions "angry", "fear", "happy" and "sad" and for **phrase one** for the song **"Comparsita".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.3093 | **0.4808** | 0.1177 | 0.2646 | 0.4619 |
| Fear | 0.1559 | **0.3734** | 0.0524 | 0.1628 | 0.2459 |
| Happy | 0.162 | **0.5115** | 0.1544 | 0.109 | 0.1554 |
| Sad | 0.1552 | **0.2923** | 0.1452 | -0.0256 | 0.1678 |

Table B.af. Correctly classified instances percentage of the **bow direction** for emotions "angry", "fear", "happy" and "sad" and for **phrase one** for the song **"Comparsita".**

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 72.33**%** | 72.33**%** | 72.33**%** | 76**%** | **76.33%** |
| Fear | **80.67%** | 74**%** | 74.33**%** | 63**%** | 72.33**%** |
| Happy | 84.33**%** | 70.67**%** | 76.67**%** | 74.67**%** | **80.67%** |
| Sad | 84.33**%** | 75.33**%** | 81**%** | 81**%** | **85%** |

Table B.ag. Correlation coefficients of the **duration ratio** and for the emotions "angry", "fear", "happy" and "sad" and for **phrase two** for the song **"Comparsita".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.4723 | 0.6248 | 0.5006 | **0.6543** | 0.633 |
| Fear | **0.7412** | 0.7073 | 0.4198 | 0.7197 | 0.6787 |
| Happy | **0.7334** | 0.712 | 0.4859 | 0.7028 | 0.6845 |
| Sad | 0.1421 | **0.431** | 0.3328 | 0.3113 | 0.1464 |

Table B.ah. Correlation coefficients of the **energy** and for the emotions "angry", "fear", "happy" and "sad" and for **phrase two** for the song **"Comparsita".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.3266 | 0.5348 | 0.1525 | **0.6307** | -0.2955 |
| Fear | 0.2146 | **0.5509** | 0.063 | 0.3929 | 0.4043 |
| Happy | 0.387 | **0.5464** | 0.2237 | 0.4243 | 0.3406 |
| Sad | 0.2889 | **0.3859** | 0.1586 | 0.2305 | 0.2999 |

Table B.ai. Correctly classified instances percentage of the **bow direction** and for the emotions "angry", "fear", "happy" and "sad" and for **phrase two** for the song **"Comparsita".**

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 66.67% | **72.33%** | 68.33% | 70% | 72% |
| Fear | **82.67%** | 70.33% | 68.67% | 62.67% | 66.67% |
| Happy | 74% | 66.67% | 72.67% | 70.67% | **76.67%** |
| Sad | **86.33%** | 76.67% | 82.33% | 84.67% | 82.67% |

Table B.aj. Correlation coefficients of the **duration ratio** and for the emotions "angry", "fear", "happy" and "sad" and for **phrase three** for the song **"Comparsita".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.8159** | 0.6836 | 0.3937 | 0.6882 | 0.7531 |
| Fear | 0.5025 | 0.8079 | **0.844** | 0.8022 | 0.6947 |
| Happy | 0.5412 | **0.7998** | 0.7845 | 0.7777 | 0.6764 |
| Sad | 0.3914 | 0.384 | **0.4722** | 0.2698 | 0.3802 |

Table B.ak. Correlation coefficients of the **energy** and for the "angry", "fear", "happy" and "sad" and for **phrase three** for the song **"Comparsita".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.6533 | **0.6856** | 0.7083 | 0.6726 | 0.4636 |
| Fear | 0.3641 | **0.6364** | 0.0973 | 0.4704 | 0.5074 |
| Happy | 0.3432 | **0.4891** | 0.195 | 0.2489 | 0.1398 |
| Sad | 0.3547 | **0.5795** | -0.0413 | 0.2442 | 0.2327 |

Table B.al. Correctly classified instances percentage of the **bow direction** and for the emotions "angry", "fear", "happy" and "sad" and for **phrase three** for the song **"Comparsita".**

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **84%** | 78.33% | **84%** | **84%** | 82% |
| Fear | **76.33%** | 68.67% | 74.67% | 74.67% | 74.67% |
| Happy | **84%** | 80.67% | 68.67% | 80.33% | 78.33% |
| Sad | 86.33% | 88.33% | **92%** | **92%** | 90% |

Table B.am. Correlation coefficients of the **duration ratio** and for the emotions "angry", "fear", "happy" and "sad" and for **phrase four** for the song **"Comparsita".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.958 | **0.9713** | 0.8236 | 0.9195 | 0.9312 |
| Fear | 0.9701 | **0.9802** | 0.7037 | 0.9533 | 0.9559 |
| Happy | 0.9223 | **0.9885** | 0.7663 | 0.9596 | 0.9804 |
| Sad | 0.5453 | **0.7915** | 0.4434 | 0.4305 | 0.5221 |

Table B.an. Correlation coefficients of the **energy** and for the emotions "angry", "fear", "happy" and "sad" and for **phrase four** for the song **"Comparsita".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.3093 | **0.4808** | 0.1177 | 0.2646 | 0.4619 |
| Fear | 0.6214 | **0.7406** | 0.6564 | 0.7025 | 0.5685 |
| Happy | 0.3073 | **0.582** | 0.1358 | 0.1584 | 0.0918 |
| Sad | 0.3925 | **0.6499** | 0.5165 | 0.4233 | 0.1968 |

Table B.ao. Correctly classified instances percentage of the **bow direction** and for the emotions "angry", "fear", "happy" and "sad" and for **phrase four** for the song **"Comparsita".**

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 89.33% | 90.67% | 87.33% | **91%** | 87.33% |
| Fear | **96.33%** | 90.67% | 94.67% | 94.67% | 94.67% |
| Happy | **88.33%** | 82.67% | 86.33% | 84.67% | 84.67% |
| Sad | 83.33% | **85.33%** | 81.33% | 81.33% | 81.33% |

## Correlation coefficients for the duration and energy models and the correctly classified instances percentage for the bow direction of the song "**Largo Invierno**":

Table B.ap. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" **when train and predict with the same data** for the song "**Largo Invierno**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.3618 | 0.4052 | **0.4262** | 0.414 | 0.3286 |
| Happy | 0.3203 | 0.2668 | **0.3598** | 0.3004 | 0.3432 |
| Sad | 0.513 | 0.4068 | 0.3973 | **0.5241** | 0.5035 |
| sweet | **0.4894** | 0.1551 | 0.4072 | 0.1331 | 0.1832 |

Table B.aq. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" wh**en train and predict with the same data** for the song "**Largo Invierno**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.5206 | **0.7028** | 0.5653 | 0.6511 | 0.5209 |
| Happy | 0.6771 | 0.6695 | 0.5093 | **0.7006** | 0.6076 |
| Sad | 0.1202 | 0.3299 | 0.2185 | **0.4192** | 0.1583 |
| sweet | 0.1113 | 0.3425 | 0.3088 | **0.3728** | 0.1286 |

Table B.ar. Correctly classified instances percentage of the **bow direction** and for the emotions "angry", "happy", "sad" and "sweet" **when train and predict with the same data** for the song "**Largo Invierno**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 57.89% | **64.47%** | 50% | 47.37% | 51.32% |
| Happy | 50.00% | 51.32% | 50% | **55.26%** | 51.32% |
| Sad | 55.26% | 61.84% | **64%** | 51.32% | 53.95% |
| sweet | **63.16%** | 55.26% | 53.95% | 48.68% | 47.37% |

Table B.as. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase one** for the song "**Largo Invierno**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.372 | 0.3625 | **0.3939** | 0.3381 | 0.2184 |
| Happy | 0.4044 | 0.2747 | 0.4167 | 0.2922 | **0.4943** |
| Sad | 0.4012 | 0.4926 | 0.5704 | **0.6055** | 0.5311 |
| sweet | **0.581** | 0.2577 | 0.415 | 0.2943 | 0.281 |

Table B.at. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase one** for the song "**Largo Invierno**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.5545 | **0.7872** | 0.6735 | 0.5966 | 0.6638 |
| Happy | 0.5349 | 0.6602 | 0.5992 | **0.6844** | 0.5526 |
| Sad | 0.3071 | **0.3671** | 0.1655 | 0.3034 | 0.1898 |
| sweet | 0.381 | **0.44** | 0.3038 | 0.4037 | 0.0567 |

Table B.au. Correctly classified instances percentage of the **bow direction** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase one** for the song "**Largo Invierno**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 62.71% | 64.41% | **67.80%** | 59.32% | 66.10% |
| Happy | 42.37% | 44.07% | 45.76% | **55.93%** | 47.46% |
| Sad | 61.02% | 55.93% | 64.41% | 61.02% | **66.10%** |
| sweet | 54.24% | **59.32%** | 49.15% | 57.63% | 50.85% |

Table B.av. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase two** for the song "**Largo Invierno**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.362 | **0.3985** | 0.2619 | 0.3432 | 0.2864 |
| Happy | 0.2122 | 0.2129 | 0.4187 | 0.2559 | **0.4281** |
| Sad | 0.4207 | 0.3673 | 0.434 | 0.4611 | **0.4878** |
| sweet | **0.2364** | 0.1164 | 0.1355 | 0.1355 | 0.1229 |

Table B.aw. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase two** for the song "**Largo Invierno**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.6667 | **0.7641** | 0.7492 | 0.7314 | 0.6415 |
| Happy | 0.5868 | 0.7179 | 0.535 | **0.7277** | 0.4782 |
| Sad | 0.1943 | 0.4808 | 0.2209 | **0.505** | 0.2148 |
| sweet | 0.1848 | **0.3485** | 0.1801 | 0.2561 | 0.1693 |

Table B.ax. Correctly classified instances percentage of the **bow direction** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase two** for the song "**Largo Invierno**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **66.67%** | 62.75% | 62.75% | 54.90% | 58.82% |
| Happy | 39.22% | 50.98% | 47.06% | 50.98% | **54.90%** |
| Sad | 49.02% | 52.94% | **54.90%** | 49.02% | 50.98% |
| sweet | 52.94% | 49.02% | 45.10% | 50.98% | **56.86%** |

Table B.ay. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase three** for the song "**Largo Invierno**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.0423 | 0.3189 | 0.1972 | **0.4376** | 0.3315 |
| Happy | 0.1583 | 0.1782 | **0.4531** | 0.0446 | 0.2073 |
| Sad | 0.2869 | -0.0203 | **0.6446** | 0.2612 | 0.4418 |
| sweet | 0.0961 | **0.2471** | 0.2253 | 0.2185 | 0.0961 |

Table B.az. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase three** for the song "**Largo Invierno**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | -0.0433 | 0.1708 | 0.0985 | 0.0829 | **0.2239** |
| Happy | -0.0022 | 0.291 | 0.2074 | 0.2211 | **0.3137** |
| Sad | 0.1964 | 0.1997 | **0.6446** | 0.2612 | 0.2416 |
| sweet | 0.1817 | **0.4225** | 0.0353 | 0.1955 | 0.3837 |

Table B.ba. Correctly classified instances percentage of the **bow direction** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase three** for the song "**Largo Invierno**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 57.41% | **61.11%** | 55.56% | 44.44% | 50.00% |
| Happy | 57.41% | 50.00% | **59.26%** | 59.26% | 53.70% |
| Sad | 62.96% | 50.00% | **62.96%** | 51.85% | 53.70% |
| sweet | **64.81%** | 48.15% | 46.30% | 42.59% | 40.74% |

Table B.bb. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase four** for the song "**Largo Invierno**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.3273 | 0.3756 | **0.4304** | 0.3761 | 0.3339 |
| Happy | **0.406** | 0.3001 | 0.2947 | 0.2855 | 0.2529 |
| Sad | 0.6797 | **0.7109** | 0.5738 | 0.6998 | 0.6638 |
| sweet | 0.311 | **0.3213** | 0.1854 | 0.1854 | 0.2521 |

Table B.bc. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase four** for the song "**Largo Invierno**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.6019 | **0.732** | 0.6155 | 0.6829 | 0.5894 |
| Happy | 0.5698 | 0.707 | 0.5774 | **0.7269** | 0.446 |
| Sad | 0.3938 | 0.271 | 0.3628 | **0.4373** | 0.3001 |
| sweet | **0.3962** | 0.2356 | 0.3481 | 0.2963 | 0.299 |

Table B.bd. Correctly classified instances percentage of the **bow direction** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase four** for the song "**Largo Invierno**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 53.13% | **62.50%** | 51.56% | 51.56% | 56.25% |
| Happy | **54.69%** | 46.88% | 40.63% | 46.88% | 46.88% |
| Sad | 51.56% | **62.50%** | 57.81% | 60.94% | 54.69% |
| sweet | 45.31% | **58.33%** | 48.44% | 50.00% | 48.44% |

Correlation coefficients for the duration and energy models and the correctly classified instances percentage for the bow direction of the song "**Por una cabeza**":

Table B.be. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" **when train and predict with the same data** for the song "**Por una cabeza**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.3 | **0.33** | 0.14 | 0.32 | 0.3 |
| Happy | 0.16 | **0.43** | 0.26 | 0.3 | 0.33 |
| Sad | **0.48** | 0.44 | 0.32 | 0.29 | 0.12 |
| sweet | 0.42 | **0.52** | 0.44 | 0.42 | 0.28 |

Table B.bf. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" **when train and predict with the same data** for the song "**Por una cabeza**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.19 | 0.07 | **0.23** | **0.23** | 0.1 |
| Happy | 0.11 | 0.11 | 0.09 | 0.28 | **0.29** |
| Sad | 0.2 | 0.04 | 0.21 | **0.27** | -0.06 |
| sweet | **0.07** | -0.08 | **0.07** | **0.07** | 0.05 |

Table B.ag. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" **when train and predict with the same data** for the song "**Por una cabeza**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Happy | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Sad | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| sweet | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |

Table B.bh. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase one** for the song "**Por una cabeza**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.17 | **0.32** | 0.19 | 0.18 | 0.16 |
| Happy | 0.24 | 0.24 | 0.22 | **0.43** | 0.26 |
| Sad | **0.56** | 0.38 | 0.42 | 0.4 | 0.29 |
| sweet | 0.38 | **0.49** | 0.4 | 0.25 | 0.17 |

Table B.bi. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase one** for the song "**Por una cabeza**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.17** | -0.18 | 0.09 | 0.05 | 0.09 |
| Happy | 0.28 | **0.33** | 0.26 | 0.19 | **0.33** |
| Sad | 0.12 | 0.06 | 0.16 | **0.2** | 0.03 |
| sweet | 0.05 | -0.03 | 0.04 | 0.01 | **0.1** |

Table B.bj. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" and for **phrase one** for the song "**Por una cabeza**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Happy | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Sad | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| sweet | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |

Table B.bk. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase two** for the song "**Por una cabeza**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.02 | **0.26** | -0.05 | 0.24 | 0.19 |
| Happy | **0.4** | 0.28 | 0.24 | 0.26 | 0.22 |
| Sad | 0.02 | **0.36** | 0.01 | 0.06 | -0.07 |
| sweet | **0.17** | 0.09 | 0.06 | 0.13 | 0.04 |

Table B.bl. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase two** for the song "**Por una cabeza**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.03 | -0.16 | **0.09** | -0.02 | -0.06 |
| Happy | **0.34** | 0.31 | 0.17 | 0.15 | 0.07 |
| Sad | 0.03 | 0.07 | -0.02 | **0.26** | 0.23 |
| sweet | **0.03** | -0.07 | -0.26 | -0.16 | -0.22 |

Table B.bm. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" and for **phrase two** for the song "**Por una cabeza**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Happy | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Sad | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| sweet | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |

Table B.bn. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase three** for the song "**Por una cabeza**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.08 | 0.05 | **0.14** | -0.01 | 0.08 |
| Happy | **0.27** | 0.01 | 0.25 | -0.01 | 0.08 |
| Sad | **0.26** | 0.18 | 0.1 | 0.06 | 0.19 |
| sweet | 0.13 | 0.17 | **0.32** | 0.12 | 0.15 |

Table B.bo. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase three** for the song "**Por una cabeza**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.19 | 0 | -0.01 | 0.25 | **0.31** |
| Happy | -0.25 | 0.18 | 0.05 | **0.28** | 0.26 |
| Sad | 0.18 | 0.15 | 0.3 | **0.35** | 0.22 |
| sweet | 0.03 | 0.29 | **0.4** | 0.12 | 0.26 |

Table B.bp. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" and for **phrase three** for the song "**Por una cabeza**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Happy | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Sad | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| sweet | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |

Table B.bq. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase four** for the song "**Por una cabeza**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.21 | **0.38** | 0.07 | 0.18 | 0.26 |
| Happy | 0.29 | 0.46 | 0.07 | **0.47** | 0.37 |
| Sad | **0.26** | 0.18 | 0.1 | 0.06 | 0.19 |
| sweet | 0.13 | 0.17 | **0.32** | 0.12 | 0.15 |

Table B.br. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase four** for the song "**Por una cabeza**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.31** | 0.1 | 0.2 | 0.2 | 0.1 |
| Happy | -0.1 | -0.07 | 0.02 | 0.14 | **0.24** |
| Sad | **0.25** | 0.21 | 0.13 | 0.17 | 0.2 |
| sweet | **0.31** | 0.1 | 0.2 | 0.2 | 0.1 |

Table B.bs. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" and for **phrase four** for the song "**Por una cabeza**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Happy | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Sad | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| sweet | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |

## Correlation coefficients for the duration and energy models and the correctly classified instances percentage for the bow direction of the song "**Primavera**":

Table B.bt. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" **when train and predict with the same data** for the song "**Primavera**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.53 | **0.61** | 0.49 | 0.48 | 0.51 |
| Happy | 0.47 | **0.65** | 0.57 | 0.51 | 0.36 |
| Sad | 0.39 | **0.56** | 0.4 | 0.43 | 0.5 |
| sweet | 0.38 | **0.46** | 0.26 | 0.22 | 0.23 |

Table B.bu. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" **when train and predict with the same data** for the song "**Primavera**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.53** | 0.26 | 0.46 | 0.42 | 0.25 |
| Happy | 0.31 | **0.55** | 0.33 | 0.42 | 0.37 |
| Sad | 0.16 | 0.36 | 0.28 | **0.55** | 0.2 |
| sweet | 0.27 | 0.2 | **0.46** | 0.4 | 0.34 |

Table B.bv. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" **when train and predict with the same data** for the song "**Primavera**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Happy | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Sad | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| sweet | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |

Table B.bw. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase one** for the song "**Primavera**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.49 | 0.52 | 0.4 | **0.57** | 0.5 |
| Happy | 0.41 | **0.7** | 0.46 | 0.53 | 0.42 |
| Sad | 0.12 | **0.53** | 0.21 | 0.53 | 0.23 |
| sweet | 0.41 | **0.48** | 0.05 | 0.45 | 0.35 |

Table B.bx. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase one** for the song "**Primavera**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.4** | 0.33 | 0.35 | 0.37 | 0.34 |
| Happy | 0.5 | 0.51 | 0.23 | 0.37 | **0.56** |
| Sad | 0.31 | 0.41 | 0.17 | **0.5** | 0.48 |
| sweet | 0.52 | 0.46 | 0.43 | 0.61 | **0.62** |

Table B.by. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" and for **phrase one** for the song "**Primavera**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Happy | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Sad | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| sweet | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |

Table B.bz. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase two** for the song "**Primavera**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.52 | 0.6 | 0.39 | 0.56 | **0.65** |
| Happy | 0.42 | **0.63** | 0.49 | 0.55 | 0.53 |
| Sad | 0.32 | 0.52 | 0.27 | **0.54** | 0.48 |
| sweet | 0.34 | 0.5 | 0.29 | **0.54** | 0.49 |

Table B.ca. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase two** for the song "**Primavera**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.66** | 0.52 | 0.5 | 0.56 | 0.58 |
| Happy | 0.28 | 0.24 | 0.12 | 0.27 | **0.55** |
| Sad | 0.33 | 0.39 | 0.26 | **0.45** | 0.28 |
| sweet | **0.52** | 0.43 | 0.42 | 0.43 | 0.4 |

Table B.cb. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" and for **phrase two** for the song "**Primavera**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Happy | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Sad | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| sweet | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |

Table B.cc. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase three** for the song "**Primavera**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.63 | **0.64** | 0.51 | 0.62 | 0.57 |
| Happy | 0.53 | 0.58 | 0.42 | **0.6** | 0.51 |
| Sad | 0.24 | **0.63** | 0.46 | 0.46 | 0.5 |
| sweet | 0.11 | 0.3 | **0.47** | 0.1 | 0.33 |

Table B.cd. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase two** for the song "**Primavera**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.47 | 0.47 | **0.59** | 0.53 | 0.51 |
| Happy | 0.53 | 0.5 | **0.65** | 0.41 | 0.62 |
| Sad | **0.47** | 0.39 | 0.33 | 0.42 | 0.32 |
| sweet | 0.43 | 0.43 | **0.53** | 0.52 | 0.51 |

Table B.ce. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" and for **phrase three** for the song "**Primavera**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Happy | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Sad | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| sweet | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |

Table B.cf. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase four** for the song "**Primavera**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.72** | 0.65 | 0.56 | 0.57 | 0.6 |
| Happy | 0.31 | 0.6 | 0.39 | **0.62** | 0.56 |
| Sad | 0.44 | 0.53 | **0.54** | 0.31 | 0.47 |
| sweet | **0.52** | 0.35 | 0.5 | 0.31 | 0.32 |

Table B.cg. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "sad" and "sweet" and for **phrase four** for the song "**Primavera**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.46** | 0.24 | 0.39 | 0.38 | 0.22 |
| Happy | 0.55 | 0.58 | 0.34 | 0.69 | **0.78** |
| Sad | 0.37 | 0.58 | 0.32 | **0.61** | 0.43 |
| sweet | 0.28 | 0.25 | 0.33 | **0.46** | 0.39 |

Table B.ch. Correctly classified instances percentage of the **bow direction** for the emotions "angry", "fear", "happy" and "sad" and for **phrase four** for the song "**Primavera**".

| **Bow direction** | Decision trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Happy | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| Sad | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |
| sweet | 100.00% | 100.00% | 100.00% | 100.00% | 100.00% |

## Correlation coefficients for the duration and energy of the song "**Boblicity":**

Table B.ci. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" **when train and predict with the same data** for the song "**Boblicity**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.44 | 0.49 | 0.45 | **0.59** | 0.39 |
| Fear | 0.55 | 0.56 | 0.5 | **0.61** | 0.45 |
| Happy | 0.79 | 0.83 | 0.68 | **0.85** | 0.76 |
| Sad | 0.59 | 0.65 | 0.54 | **0.66** | 0.56 |

Table B.cj. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" **when train and predict with the same data** for the song "**Boblicity**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.61** | 0.44 | 0.52 | 0.53 | 0.42 |
| Fear | 0.58 | 0.65 | 0.65 | **0.67** | 0.6 |
| Happy | 0.47 | 0.44 | 0.48 | **0.55** | 0.52 |
| Sad | **0.45** | 0.34 | 0.37 | 0.42 | 0.3 |

Table B.ck. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase one** for the song "**Boblicity**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.41 | 0.36 | 0.36 | **0.41** | 0.29 |
| Fear | 0.51 | 0.46 | 0.48 | **0.64** | 0.44 |
| Happy | **0.81** | 0.76 | 0.7 | 0.76 | 0.75 |
| Sad | **0.61** | 0.52 | 0.49 | 0.45 | 0.43 |

Table B.cl. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase one** for the song "**Boblicity**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.46 | 0.48 | **0.63** | 0.46 | 0.42 |
| Fear | 0.52 | 0.63 | **0.65** | 0.6 | 0.54 |
| Happy | 0.47 | 0.36 | **0.47** | 0.36 | 0.45 |
| Sad | 0.21 | 0.1 | 0.21 | **0.28** | 0.12 |

Table B.cm. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase two** for the song "**Boblicity**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.63** | 0.47 | 0.57 | 0.55 | 0.42 |
| Fear | 0.42 | 0.43 | 0.34 | **0.49** | 0.32 |
| Happy | 0.68 | **0.74** | 0.67 | 0.69 | 0.72 |
| Sad | 0.57 | 0.48 | **0.59** | 0.48 | 0.45 |

Table B.cn. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase two** for the song "**Boblicity**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.51 | 0.48 | **0.53** | 0.45 | 0.35 |
| Fear | 0.63 | 0.57 | 0.63 | **0.64** | 0.54 |
| Happy | 0.38 | 0.24 | **0.49** | 0.23 | 0.3 |
| Sad | 0.26 | 0.27 | **0.35** | 0.21 | **0.35** |

Table B.co. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase three** for the song "**Boblicity**".

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.56** | 0.52 | 0.44 | 0.48 | 0.43 |
| Fear | 0.52 | 0.6 | 0.49 | **0.67** | 0.52 |
| Happy | 0.82 | **0.87** | 0.7 | 0.79 | 0.79 |
| Sad | 0.59 | **0.78** | 0.57 | 0.76 | 0.55 |

Table B.cp. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase three** for the song "**Boblicity**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.47 | **0.61** | 0.58 | 0.54 | 0.5 |
| Fear | 0.61 | 0.73 | 0.67 | **0.76** | 0.72 |
| Happy | 0.57 | **0.65** | 0.56 | 0.64 | 0.59 |
| Sad | **0.49** | 0.44 | 0.35 | 0.45 | 0.3 |

Table B.cq. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase four** for the song "**Boblicity**".

| **Duration** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.47 | **0.6** | 0.48 | 0.54 | 0.41 |
| Fear | **0.62** | 0.6 | 0.5 | 0.59 | 0.6 |
| Happy | 0.65 | **0.88** | 0.66 | 0.82 | 0.8 |
| Sad | 0.56 | **0.63** | 0.58 | 0.59 | 0.56 |

Table B.cr. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase four** for the song "**Boblicity**".

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.51 | 0.52 | 0.56 | **0.58** | 0.56 |
| Fear | 0.57 | 0.58 | 0.68 | 0.7 | **0.75** |
| Happy | 0.46 | 0.58 | 0.44 | **0.59** | 0.58 |
| Sad | **0.52** | 0.48 | 0.49 | 0.5 | 0.5 |

## Correlation coefficients for the duration and energy of the song "**How deep is the ocean":**

Table B.cs. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" **when train and predict with the same data** for the song "**How deep is the ocean".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.5 | 0.35 | **0.51** | 0.36 | 0.46 |
| Fear | **0.48** | 0.29 | 0.4 | 0.41 | 0.23 |
| Happy | 0.44 | 0.34 | **0.52** | 0.36 | 0.23 |
| Sad | **0.58** | 0.37 | 0.56 | 0.3 | 0.4 |

Table B.ct. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" **when train and predict with the same data** for the song "**How deep is the ocean".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.67** | 0.46 | 0.41 | 0.32 | 0.57 |
| Fear | 0.4 | 0.39 | **0.52** | 0.38 | 0.24 |
| Happy | 0.46 | 0.29 | **0.48** | 0.31 | 0.2 |
| Sad | **0.42** | 0.2 | 0.37 | 0.33 | 0.24 |

Table B.cu. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase one** for the song "**How deep is the ocean".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.44 | **0.52** | 0.39 | 0.4 | 0.45 |
| Fear | **0.62** | 0.36 | 0.49 | 0.4 | 0.52 |
| Happy | 0.29 | 0.36 | 0.23 | **0.38** | 0.35 |
| Sad | 0.3 | 0.38 | **0.45** | 0.21 | 0.18 |

Table B.cv. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase one** for the song "**How deep is the ocean".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.6** | 0.45 | 0.38 | 0.36 | 0.43 |
| Fear | 0.31 | 0.2 | **0.39** | 0.3 | 0.19 |
| Happy | **0.41** | 0.06 | 0.26 | 0.14 | 0.19 |
| Sad | **0.35** | 0.13 | 0.02 | 0 | 0.21 |

Table B.cw. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase two** for the song "**How deep is the ocean".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.68 | **0.72** | 0.63 | 0.51 | 0.3 |
| Fear | **0.47** | 0.37 | 0.46 | 0.46 | 0.33 |
| Happy | 0.23 | 0.48 | **0.53** | 0.34 | 0.34 |
| Sad | **0.6** | 0.4 | 0.58 | 0.25 | 0.32 |

Table B.cx. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase two** for the song "**How deep is the ocean".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.57 | 0.64 | **0.68** | 0.57 | 0.59 |
| Fear | **0.46** | 0.36 | 0.34 | 0.31 | 0.09 |
| Happy | 0.35 | 0.26 | 0.35 | **0.38** | 0.12 |
| Sad | **0.53** | 0.16 | **0.53** | 0.24 | 0.42 |

Table B.cy. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase three** for the song "**How deep is the ocean".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.26** | 0.24 | 0.16 | 0 | 0.05 |
| Fear | 0.51 | 0.35 | **0.53** | 0.26 | 0.3 |
| Happy | 0.18 | 0.26 | **0.39** | 0.31 | 0.29 |
| Sad | 0.32 | 0.23 | **0.35** | 0.07 | 0.16 |

Table B.cz. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase three** for the song "**How deep is the ocean".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.77** | 0.55 | 0.51 | 0.54 | 0.55 |
| Fear | **0.66** | 0.51 | 0.63 | 0.58 | 0.6 |
| Happy | **0.28** | 0.05 | 0.15 | 0.04 | 0.08 |
| Sad | **0.41** | 0.3 | 0.25 | 0.24 | 0.36 |

Table B.da. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase four** for the song "**How deep is the ocean".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.4 | 0.3 | **0.45** | 0.32 | 0.22 |
| Fear | **0.61** | 0.49 | 0.58 | 0.57 | 0.57 |
| Happy | 0.35 | 0.33 | **0.54** | 0.4 | 0.32 |
| Sad | 0.53 | 0.34 | **0.65** | 0.2 | 0.37 |

Table B.db. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase four** for the song "**How deep is the ocean".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.16 | 0.13 | **0.39** | 0.31 | 0.19 |
| Fear | 0.37 | 0.13 | **0.5** | 0.22 | 0.29 |
| Happy | **0.68** | 0.38 | 0.6 | 0.46 | 0.4 |
| Sad | **0.27** | 0.03 | 0.16 | 0.22 | -0.05 |

## Correlation coefficients for the duration and energy of the song "**Lullaby of birdland":**

Table B.dc. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" **when train and predict with the same data** for the song "**Lullaby of birdland".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.4127** | 0.3497 | 0.3524 | 0.4093 | 0.2081 |
| Fear | 0.0234 | 0.1393 | 0.231 | 0.0608 | **0.2465** |
| Happy | 0.5559 | 0.2397 | **0.6027** | 0.49 | 0.5647 |
| Sad | 0.2001 | 0.0985 | 0.2366 | 0.1271 | **0.3485** |

Table B.dd. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" **when train and predict with the same data** for the song "**Lullaby of birdland".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.1412 | **0.6811** | 0.2701 | 0.3272 | 0.1711 |
| Fear | 0.4698 | 0.6798 | 0.6201 | 0.6458 | **0.7512** |
| Happy | 0.232 | 0.0939 | **0.2784** | 0.249 | 0.0683 |
| Sad | 0.3587 | 0.276 | 0.4004 | **0.4389** | 0.2631 |

Table B.de. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase one** for the song "**Lullaby of birdland".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | **0.4577** | 0.3418 | 0.2748 | 0.3885 | 0.373 |
| Fear | 0.0871 | 0.0535 | 0.1124 | **0.2127** | 0.1597 |
| Happy | **0.5585** | 0.212 | 0.5386 | 0.534 | 0.4869 |
| Sad | 0.074 | 0.0278 | 0.0422 | **0.1849** | 0.1848 |

Table B.df. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase one** for the song "**Lullaby of birdland".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.1773 | 0.1443 | 0.3912 | **0.4465** | 0.1713 |
| Fear | 0.4375 | 0.5868 | 0.6197 | 0.6632 | **0.6796** |
| Happy | 0.2367 | 0.0032 | 0.2999 | **0.3138** | 0.1654 |
| Sad | 0.3233 | 0.256 | **0.4987** | 0.4131 | 0.3616 |

Table B.dg. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase two** for the song "**Lullaby of birdland".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.3692 | 0.3502 | 0.3323 | 0.4557 | **0.456** |
| Fear | 0.1596 | 0.136 | **0.2151** | 0.0792 | 0.1383 |
| Happy | **0.5585** | 0.212 | 0.5386 | 0.534 | 0.4869 |
| Sad | 0.1029 | 0.0765 | **0.1981** | 0.155 | 0.1534 |

Table B.dh. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase two** for the song "**Lullaby of birdland".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.0998 | 0.0914 | 0.2031 | **0.3045** | 0.1768 |
| Fear | 0.1231 | 0.4915 | **0.526** | 0.3847 | 0.3968 |
| Happy | 0.1909 | -0.011 | 0.2738 | **0.2951** | 0.1402 |
| Sad | 0.2493 | 0.2232 | 0.3456 | **0.411** | 0.2992 |

Table B.di. Correlation coefficients of the **duration ratio** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase three** for the song "**Lullaby of birdland".**

| **Duration ratio** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.4238 | 0.3002 | 0.392 | **0.4757** | 0.3486 |
| Fear | 0.1076 | **0.1382** | 0.0704 | 0.1716 | 0.1215 |
| Happy | **0.5585** | 0.212 | 0.5386 | 0.534 | 0.4869 |
| Sad | 0.1713 | 0.0527 | 0.1727 | 0.1191 | **0.3277** |

Table B.dj. Correlation coefficients of the **energy** and for the emotions "angry", "happy", "fear" and "sad" and for **phrase three** for the song "**Lullaby of birdland".**

| **Energy** | Regression trees | kNN (k=1) | SVM linear kernel | SVM 2nd degree polynomial | ANN (one hidden layer) |
|---|---|---|---|---|---|
| Angry | 0.4682 | 0.3514 | **0.5486** | 0.3754 | 0.3089 |
| Fear | 0.5148 | **0.6796** | 0.615 | 0.6419 | 0.6351 |
| Happy | 0.3901 | 0.1789 | **0.4564** | 0.2471 | 0.2372 |
| Sad | **0.6522** | 0.4859 | 0.5745 | 0.3976 | 0.3794 |