

BACHELOR THESIS

# **Emotion recognition based on the speech, using a Naive Bayes Classifier**

Submitted at the  
Institute of Computer Technology,  
TU Wien  
in partial fulfillment of the requirements for the degree of  
Telematics Engineering

under supervision of

Nima Taherinejad  
Institute number: 384  
Institute for Computer Technology  
and  
Antonio Bonafonte  
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

by

Angel Urbano Romeu  
Matr.Nr. 1529643  
43764, Tarragona, Spain

30.06.2016

## **Abstract**

Speech emotion recognition is one of the major challenges in speech processing. Besides facial expressions or gestures, speech has proven as one of the most promising modalities for the automatic emotion recognition. To identify the emotions from the speech signal, many systems have been developed. This project presents the results from the application of a Naive Bayes classifier over various types of features. Automatic detection of emotions has been evaluated using standard Mel-Frequency Cepstral Coefficients (MFCCs), and pitch related features extracted from a speech segment. These segments contain a set of recorded sentences by actors and actresses who express different emotions.

The classification performance is based on extracted features. The best results have approximately 78% of accuracy using proper layers and weights in the classifier. Classifying emotions with Naive Bayes provides quick probabilistic results and performs better than more sophisticated classifiers.

# Table of contents

1. Introduction .....	1
1.1 Motivation .....	1
1.2 Problem Statement .....	1
1.3 Task Setting .....	1
1.4 Methodology .....	1
2. State of the Art and Related Works .....	2
2.1 Databases review .....	4
2.2 Feature extraction .....	4
2.3 Classifier .....	6
2.3.1 SVM .....	6
2.3.2 KNN .....	7
2.3.3 GMM .....	7
2.3.4 Naïve Bayes (NB) .....	8
2.3.5 Results for emotion recognition .....	8
3. Model and Concepts .....	11
3.1 Database of Emotional Speech .....	11
3.2 Feature extraction .....	11
3.2.1 Pitch and related features .....	12
3.2.2 Mel Frequency Cepstral Coefficients (MFCC) and related features .....	15
3.3 Classification .....	17
3.3.1 How Naïve Bayes works .....	17
3.3.2 Classification improvements .....	19
4. Implementation and Results .....	21
4.1 Feature extraction using OpenSMILE and CSV database .....	21
4.2 Classification (Training and Testing) .....	23
4.3 Simulation Results .....	27

5. Conclusions and Future Works .....	29
5.1 Future work .....	29
6. References .....	30
Declaration .....	1

# Abbreviations

CSV. Comma Separated Values

DCT. Discrete Cosine Transform

MFCC. Mel Frequency Cepstral Coefficients

LPCC. Linear Prediction Cepstral Coefficients

SVM. Support Vector Machine

GMM. Gaussian Mixture Model

KNN. K-nearest neighbors

NB. Naïve Bayes

PDA. Pitch Detection Algorithm

WAV. Waveform Audio Format

# **1. Introduction**

Traditionally, emotions in machines have had no role in their internal decision systems. However, recent discoveries in neurosciences, together with emotional intelligence have led to the emergence of a new framework “Affective Computing”, according to which, the aim is to build machines that recognize, communicate and respond to user emotions indicators [1].

## **1.1 Motivation**

Emotion recognition through speech processing is a discipline that is increasing the interest in the human-machine interaction. The aim is the automatic identification of the emotional state of humans from their voice [2]. The actual user emotion may help a system track the user’s behaviour by adapting to his inner state. Among others like facial expressions or mimic, recognizing emotions by the speech is one of the most promising and established modalities for the recognition.

## **1.2 Problem Statement**

The emotion recognition systems have the aim of recognizing emotions, in this case, from the speech. The problems introduced to these systems are:

How the emotions are presented inside an audio signal? How can a classifier use labelled samples to classify the emotion of a new one?

## **1.3 Task Setting**

The task of this work is to develop a complete machine learning system able to decide the emotion of a new audio signal where the emotion is unknown.

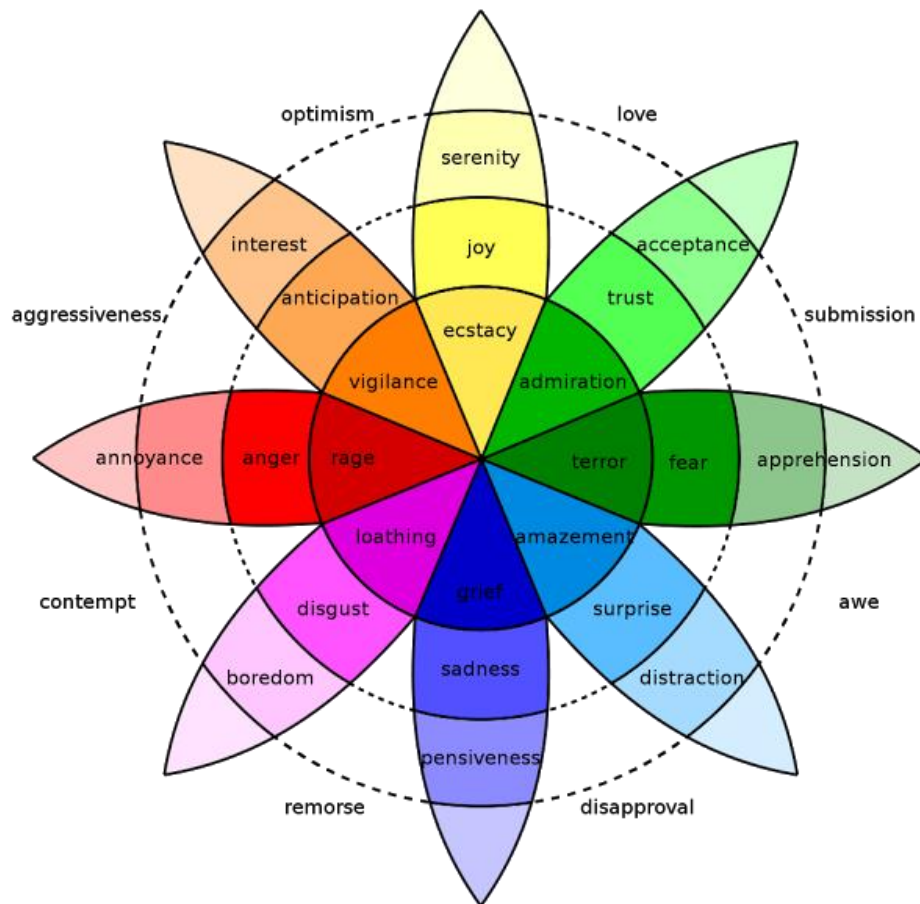
## **1.4 Methodology**

Machine learning systems need three main steps, the database, the classifier and the features. To create a system capable to learn and predict emotions, firstly a database is selected, then comes the extraction of pitch and MFCCs related features and last but not least, the development of the classifier, which is implemented and finally tested.

## 2. State of the Art and Related Works

Emotion is an important factor in communication. This is why emotion recognition from speech has emerged as an important research area in the recent past [3]. Speech is a complex signal that contains lots of information about the message, speaker, language and emotions. Emotions make speech more expressive and effective. Emotion recognition means detection of the emotional state of human through features extracted from his or her recorded voice signal. Emotion recognition through Speech is particularly useful for applications in the field of human-machine interaction in order to create better interfaces [4]. The emotion recognition enables future interaction with computers or robots, which may empathise with persons or help people with communication problems and detection for psychiatric diagnosis, among other applications [5] [6] [7] [8].

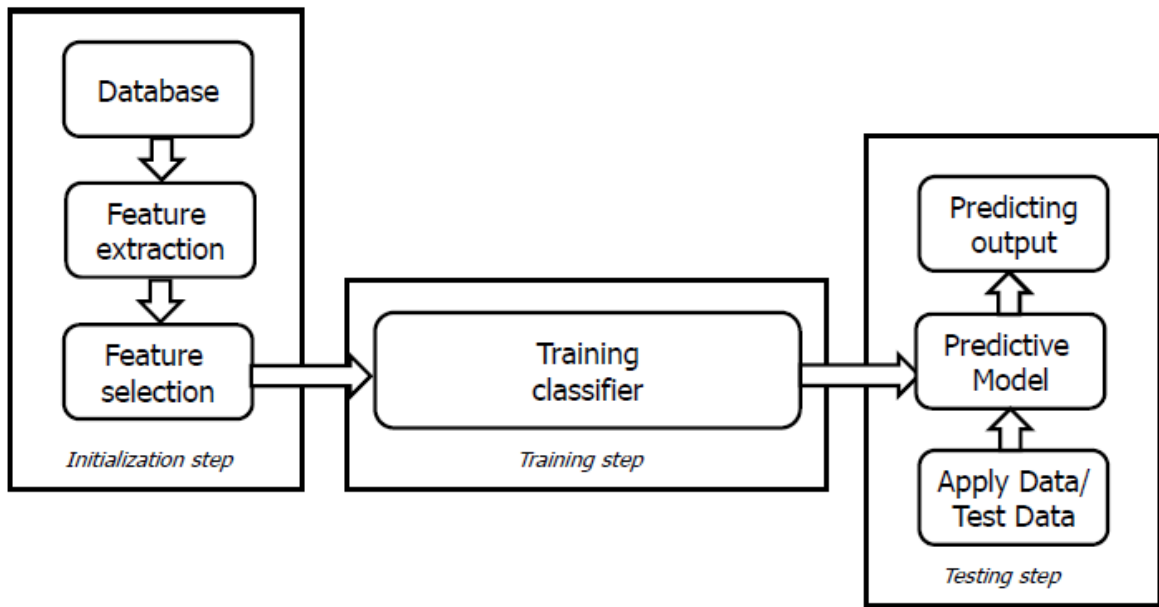
Robert Plutchik [9] was professor emeritus of Medicine and also a Psychologist, who researched, among others, about the field of emotions. He created a wheel of emotions (Figure 1), used to illustrate different emotions and describe how emotions were related. This wheel has been followed to try to understand how to discern between emotions. Try to distinguish between a few emotions using the speech can be done with the techniques that we have nowadays, but the problem appears when we try to develop software that try to distinguish between more emotions. This is why most of the developed software of emotion recognition through the speech are capable to discern well between about a maximum of 8 emotions [9], more commonly 6 emotions which are Joy, Anger, Sadness, Fear and Disgust [10].



**Figure 1: Emotions wheel (Robert Plutchik)**

To recognize these mentioned emotions, 3 steps are followed. Following the line of the machine learning systems (Figure 2), first the system needs to deal with data speech collections. Next is the features extraction where a set of audio features are extracted from the each one of the segments that compose the speech collections and finally conclude with the classification algorithm.





**Figure 2: Emotion recognition system**

## 2.1 Databases review

An important issue to be considered in the emotional speech systems is the quality of the databases used to develop them. Speech corpora can be divided into 3 types as follows:

1. Simulated based emotional speech database. The database is collected from actors that express natural sentences in different emotions [11].
2. Induced emotional speech database. Database collected by simulating artificial emotional situations [12].
3. Natural emotional speech database. Database collected from natural (not actors/actresses) sources. These kinds of databases are the worst to recognize emotions because are the most difficult in some cases.

Humans cannot easily classify natural emotions sometimes, therefore, it is difficult to expect that machines to perform a correct classification at all times.

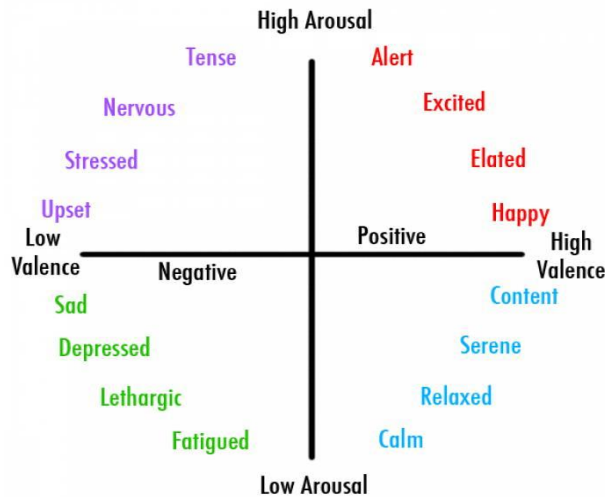
The databases for the emotion recognition by the speech are mostly labelled, which means that all the files are identified with the emotion expressed in them. This will be very useful for the machine learning classifier.

Prominent examples for acted databases are the Berlin database of emotional speech and Danish emotional speech corpus. More databases can be found in almost every language and there are studies that show the pros and cons of each one of them [13].

## 2.2 Feature extraction

Choosing suitable features for developing any machine learning systems is a crucial decision. In speech systems, the features are to be chosen to represent intended information. To classify emotions, there are two approaches about the features, the first one is the classification of emotions in categories. The second idea is to classify emotions in a coordinate system with 2 dimensions. Often arousal and valence are used as the two dimensions. Most of papers dealing with emotion

recognition in speech use categories to distinguish between emotions. From the categories extracted from small frames of the utterance, global features such as statistics can be obtained from the whole speech utterance [14].



**Figure 3: Emotion classification by arousal and valence**

The most used features are spectral and prosodic related features [15]. Speech segment of 20-30ms length will be used to extract spectral features. Vocal tract characteristics are well reflected in frequency domain analysis of speech signal; Fourier transform of speech frame gives short time spectrum. The cepstrum of a speech frame is obtained by taking the Fourier transform on log magnitude spectrum.

Statistics of the 13 first MFCCs (Mel frequency cepstral coefficients) and the LPCCs (Linear prediction cepstral coefficients) are the features derived from the cepstral domain. They can be called spectral features or vocal tract features, and are obtained from computing statistics as mean, variance, standard deviation, quartiles, range, etc. Same statistics are usually computed from the prosodic category. The prosodic related features come from the computation of the pitch, energy and intensity from the same speech segments of 20-30ms length.

There are also more features that can be obtained from the speech signal. Duration or timing features are obtained from the computation of the duration of voiced and unvoiced segments and the ratio between them [16].

## 2.3 Classifier

The last step consists on how to use the obtained features in order to determine the emotion of a new sample. Here makes its appearance into the scene, what in machine learning is called classifier. A classifier is a mapping from unlabeled instances to (discrete) classes. Classifiers have a form (e.g., decision tree) plus an interpretation procedure (including how to handle unknowns, etc.). Some classifiers also provide probability estimates (scores), which can be thresholded to yield a discrete class decision thereby taking into account a utility function [17].

To perform emotion recognition from speech various types of classifiers are used. Each classifier achieves some accuracy of results, which depends on several factors. The success of classifier is directly dependent on the data. This is derived from the fact that the accuracy varies with the data character such as the quantity and density distribution of each class [18]. Appropriate choice of parameters has a considerable effect on the accuracy of these classifiers. The most common classifiers used in the emotion recognition research are the ones described in the following subsections.

### 2.3.1 SVM

Support vector machines (SVMs) are a set of supervised learning algorithms used for classification. These algorithms' operation is based on finding the hyperplane that has the largest distance to the nearest training-data point of any class (called margin), because the larger the margin the lower the error of the classifier. These classifiers are very effective in high dimensional spaces. They use a subset of training points in the decision function called support vectors, so they are also memory efficient and, furthermore, different kernel functions can be specified for the decision function. The main disadvantage of SVMs is that if the number of features is much greater than the number of samples, the method is likely to give poor performances. In the figure below the support vector machine performance with and without kernel is shown.

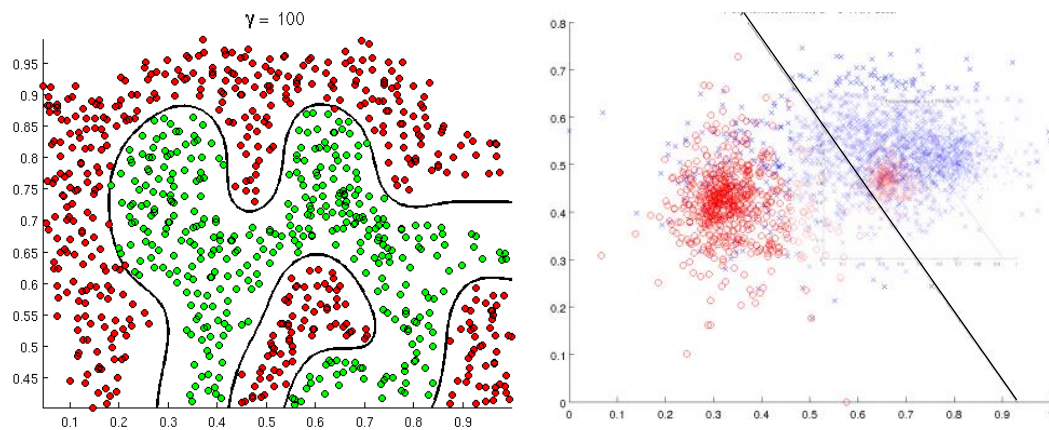
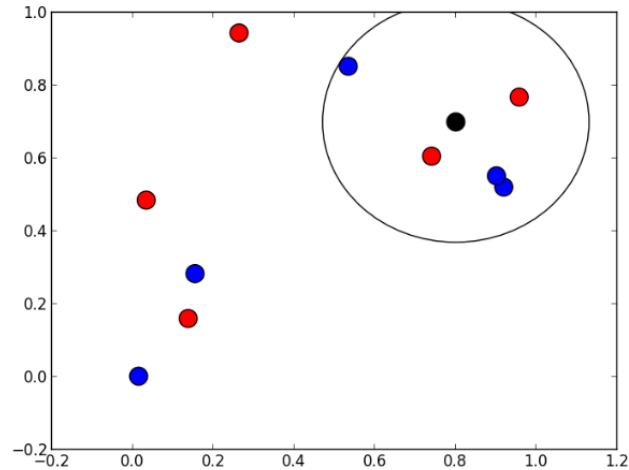


Figure 4: SVM performance with (left, using Gaussian kernel) and without kernel (right)

### 2.3.2 KNN

The k-NN (k-nearest neighbours) is a classification method that uses training data every time there is a new sample to be classified. The training samples are vectors in a multidimensional feature space and each one with a class label and the classifier will find a number of neighbours to define the class of the new sample. A constant named  $k$  is defined which is going to be the number of neighbours samples. Let's suppose the example below:



**Figure 5: k-nearest neighbours example with two classes (blue and red) and  $k=5$**

In this example,  $k=5$  which means that the classifier will search the 5 points nearby of the new sample (the black dot) and it will classify the new sample by assigning the label of the class which is most frequent among the  $k$  training samples nearest to that query point, blue in this case because there are more blue labelled samples in the five nearest neighbours than red ones.

This classifier works very well with small and large sets of data and the cost of the learning process is zero, it is also very robust to noisy training data. The main problem is that is computationally expensive to find the  $k$  nearest neighbours when the dataset is large and the testing step is very slow because it needs to compute distance of each query instance to all training samples, which means that for live applications (emotion recognition) it is not as useful.

### 2.3.3 GMM

Gaussian Mixture Models (GMMs) are among the most statistically mature methods for clustering or unsupervised learning. A Gaussian mixture model provides a good approximation of the originally observed feature probability density functions by a mixture of weighted Gaussians. Each emotion is modelled in one GMM. The decision is made for the maximum likelihood model. There are results in [19] where the authors concluded that using GMMs is a feasible technique for emotion classification.

### 2.3.4 Naïve Bayes (NB)

Naïve Bayes classifier is based on the so-called Bayesian theorem with the *naïve* assumption of independence between every pair of features. This classifier in spite of the apparently over-simplified assumptions, has worked quite well in many real-world situations. It is very fast and has a good performance, better in some cases than more sophisticated methods. The Naïve Bayes model with Gaussian is equivalent to a mixture of Gaussians (GMM) with diagonal covariance matrices. The main advantages of this classifier are the conditional independence assumption, which helps to obtain a quick classification, and the probabilistic hypotheses (results obtained as probabilities of belonging of each class).

### 2.3.5 Results for emotion recognition

All the mentioned classifiers are often used by researchers on emotion recognition from the speech. The following tables show the classification accuracy and the confusion matrixes of every one of the classifiers obtained from some of the sources [18, 20-27].

#### SVM

The overall accuracy of SVM is around 70-80% in the papers found using Emo-DB and similar. The table below shows a confusion matrix [22] of emotion recognition using the speech, classifying 5 emotions using MFCC features.

Classified as →	Angry	Happy	Sad	Neutral	Fear
Angry	<b>71</b>	14	0	0	14
Happy	0	<b>57</b>	14	0	28
Sad	0	0	<b>71</b>	28	0
Neutral	0	0	25	<b>75</b>	0
Fear	22	14	0	0	<b>63</b>

**Table 1: Confusion matrix using MFCC features with SVM classifier**

## KNN

The KNN classifier is not the most used one but it obtains good results in some cases. In the paper [26] the Berlin emotional database is used and the confusion matrix that has been obtained using the KNN classifier and some MFCC and pitch features to classify 6 emotions is the inserted in Table 2.

Classified as →	Angry	Happy	Sad	Neutral	Surprise	Fear
Angry	<b>72</b>	0	0	28	0	0
Happy	0	<b>90</b>	0	10	0	0
Sad	12	22	<b>44</b>	0	0	22
Neutral	28	0	0	<b>54</b>	4	14
Surprise	50	0	0	0	<b>25</b>	25
Fear	41	0	25	0	0	<b>34</b>

**Table 2: Confusion matrix of KNN classifier**

In this case the results are not very good (approximately 50% of accuracy), but in other papers [28], the results can reach 70-80% of accuracy using less emotions and different features.

## GMM

GMM is a classifier that has been increasingly used in emotion recognition systems during the last years. The results obtained by using this classifier are good but, in most of the cases, its average accuracy is below the SVM and Naïve Bayes classifiers. In the paper [18], a self-obtained corpora is used to compare the GMM with different classifiers using different sets of features to classify 4 emotions. The average results obtained for the GMM classifier are:

Classified as →	Neutral	Happy	Sad	Angry
Neutral	<b>58</b>	11	20	12
Happy	11	<b>73</b>	3	13
Sad	21	3	<b>72</b>	4
Angry	8	13	5	<b>72</b>

**Table 3: Confusion matrix of GMM classifier**

**NB**

In the paper [27] the Naïve Bayes is used to classify emotions. The database used is the Berlin Emotional Database and the average accuracy in this case is 77% using a set of 300 features. The confusion matrix obtained is shown below.

Classified as →	Angry	Boredom	Disgust	Fear	Happiness	Neutral	Sadness
Angry	<b>82</b>	0	0	0	18	0	0
Boredom	0	<b>77</b>	14	0	0	5	4
Disgust	4	2	<b>82</b>	2	2	6	0
Fear	10	1	3	<b>66</b>	4	11	1
Happiness	31	0	0	5	<b>62</b>	3	0
Neutral	0	9	5	5	5	<b>74</b>	1
Sadness	0	5	0	0	0	2	<b>93</b>

**Table 4: Confusion matrix of NB classifier**

## 3. Model and Concepts

As it has been mentioned before, in each pattern recognition system there are 3 basic parts that need to be used in order to solve any problem: the database, the features to extract and the classifier.

### 3.1 Database of Emotional Speech

There are many databases cited previously which are useful and open and they can be found in almost every language. In this project, the Berlin Emotional database [29] is used as a database for experiments, which contains 535 utterances of 10 actors (5 male, 5 female) simulating 7 emotional states. The emotions are anger (Wut), boredom (Langeweile), fear (Angust), sadness (Trauer), neutral, happiness (Freude) and disgust (Ekel). In this project, the emotions used are only five of them; anger, fear, sadness, neutral and happiness.

The audio files are sampled at 16kHz using the WAV file format, which admits different sample velocities and different resolutions. This format is uncompressed, which means that doesn't have losses which is an advantage for the purpose of this project.

In this project, the database will be divided in Test database and Train database. The train database will contain the 70% of the samples of the whole Berlin Emotional database and it will be used to train the classifier. The test database will contain the other 30% and it will be used for testing our classifier and obtaining results with different parameters and change them and select the best ones.

### 3.2 Feature extraction

Once the database has been selected, we need to transform the waveform (WAV) data format to speech feature numbers and statistical values that should be useful for the emotion recognition of the speaker. Any emotion from the speaker's speech is represented by a large number of parameters which are contained in the speech and the changes in these parameters will result in corresponding change in emotions. These parameters are called features, which in pattern recognition and machine learning are described as individual measurable properties of a phenomenon being observed [30].

There are many features to extract from an audio file, for this project and alike the experts are still debating on which ones are the best for every specific case of speech signal processing, but what they do know is that the most useful and used ones are the MFCC and the prosody related features [31].



The input signal in the aforementioned database is sampled at 16 kHz and converted into windows with overlapping frames, in this case the windows will be about 25ms each one. The first stage of the algorithm is to extract the features for each of these windows in the whole utterance.

Secondly, the features need to be saved with the data labeled in a format easy to access. There are various types of formats but the most used one in the machine learning cases is the comma-separated values (CSV) format. The file created contains one row for each WAV file, the rows have the features of that sample separated by commas and on the last position there is the label indicating its emotion. An example of a row is:

420; 300; -2.345; 1432; 23.33; 459.9; -0.0564; 12.354; 5.89; 1

Feature1;feature2;feature3;feature4;...;emotion number

### 3.2.1 Pitch and related features

The first extracted features are the pitch related features. The pitch is the distinctive quality of a sound, dependent primarily on the frequency of the sound waves produced by its source, which can be computed by the pitch detection algorithm (PDA) [32] implemented in software called openSMILE [33]. Once the pitch of every frame of the signal has been computed, the next step has been to use that values to compose the features. The features obtained are the following:

- Extremes: Maximum.
- Moments: Variance, Standard deviation, Skewness, Kurtosis and Mean.
- Percentiles: 1st, 2nd and 3rd quartile.

Computing the means of each feature for each emotion we can see the features that characterize the emotions and the differences between them on the tables 5 and 6.

Emotion	Maximum	Variance	Standard deviation	Skewness
Anger	422	14215	117	0.044
Boredom	362	7290	82	1.347
Disgust	389	11284	104	0.586
Fear	410	11567	106	0.634
Happiness	417	14775	119	0.074
Sadness	406	7283	83	2.353
Neutral	370	7359	82	1.629

**Table 5: Means of pitch related features for each emotion**

Emotion	Kurtosis	Mean	Quartile 1	Quartile 2	Quartile 3
Anger	2.09	152	32	164	248
Boredom	6.35	76	9	67	103
Disgust	2.48	100	0	78	193
Fear	3.21	112	7	105	184
Happiness	2.10	155	37	159	250
Sadness	10.37	47	0	11	68
Neutral	8.02	69	0.96	62	101

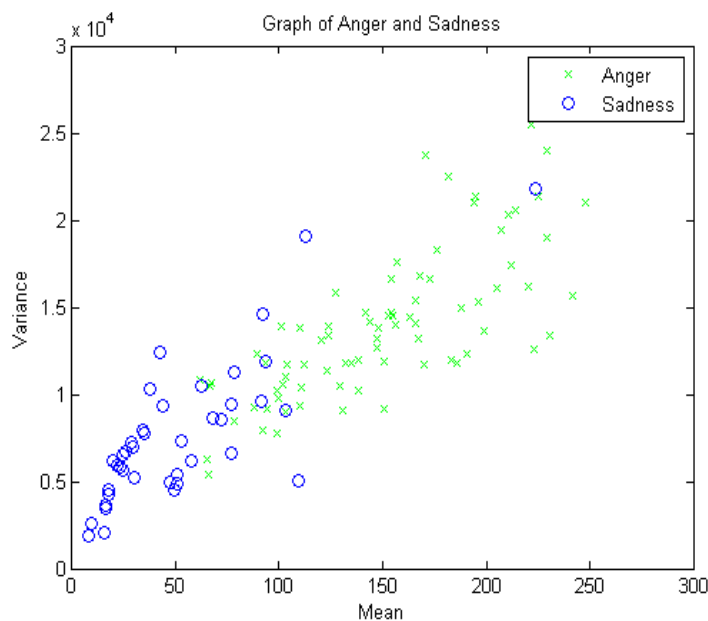
**Table 6: Means of pitch related features for each emotion**

Having a look into the tables we can see how two of the pitch related features, in this case the maximum and the mean, are higher for the Anger and Happiness and lower for the Boredom, Sadness and Neutral. Most of the features are almost the same between Anger and Happiness and between Boredom and Neutral. That is to say, there are features that distinguish more between certain emotions than others.

The next figures, Figure 6 and Figure 7, will show two emotions, Sadness and Anger, which are two distinguishable emotions bearing in mind the last tables.

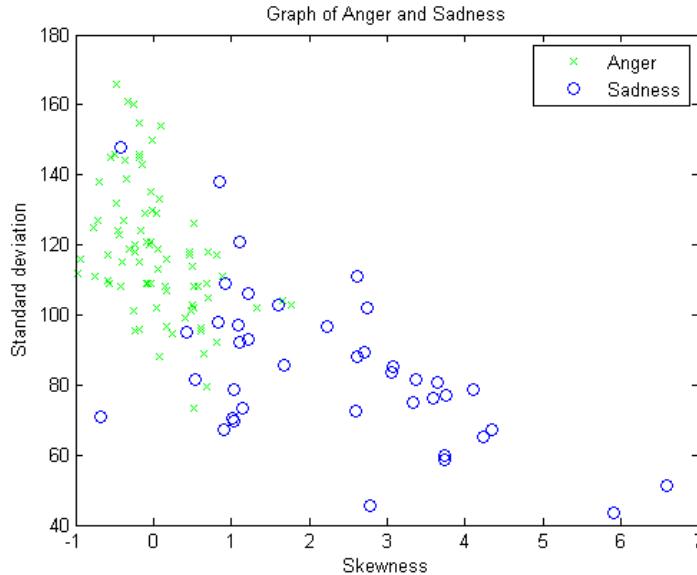
The

Figure 6 shows a 2D plot of the features Mean and Variance. Each point is a sample of a specific record from the database and each color represent one emotion of the two. The features are uncorrelated which means that are good features for the classifier.



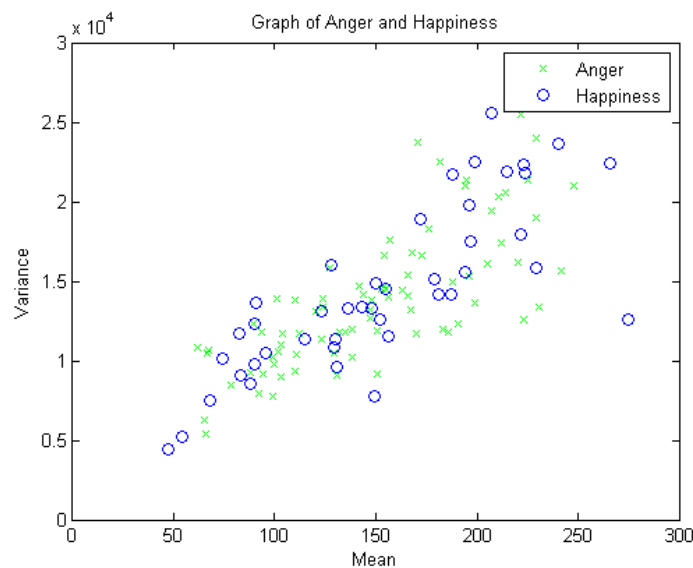
**Figure 6: Graph of Sadness and Anger using two pitch related features (Mean and Variance)**

The Figure 7 shows two different features of the same two emotions, but in this case the features selected are the Standard deviation and Skewness because looking into the means table, they are the ones more similar.



**Figure 7: Graph of Sadness and Anger using two pitch related features (Skewness and Standard deviation)**

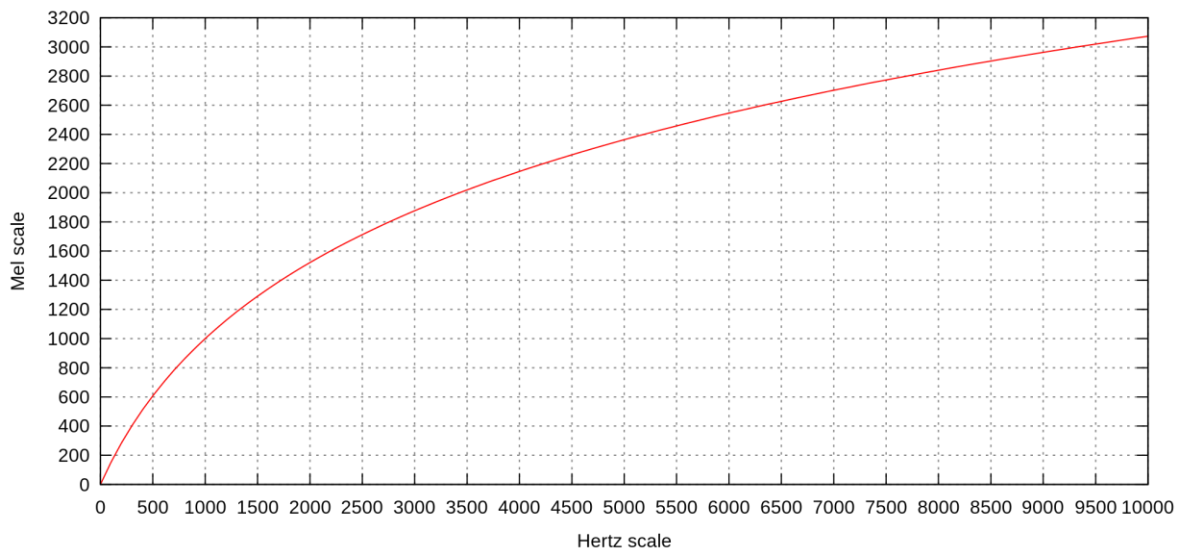
As Figure 7, even the most similar pitch related features are useful if the problem was for 2 emotions. Nevertheless, Figure 8 shows what happens if two emotions have very similar pitch related features. Since the features Mean and Variance obtained from the pitch are very similar in the case of Anger and Happiness, they don't contribute anything about the pair of emotions. In this case, the classifier wouldn't be able to perform and classify these emotions well; using the pitch related features because it would detect one emotion instead of the other most of the times.



**Figure 8: Graph of Happiness and Anger using two pitch related features (Variance and Median)**

### 3.2.2 Mel Frequency Cepstral Coefficients (MFCC) and related features

The Mel Frequency Cepstral Coefficients (MFCCs) are commonly used in sound processing, for speech recognition, emotion recognition, music modelling, etc. They were introduced by Davis and Mermelstein [34] in the 1980's and have been state of the art since then. The Mel Scale is based on a mapping between actual frequency and perceived pitch as apparently the human auditory system does not perceive pitch in a linear manner.



**Figure 9: Mel Scale**

The MFCCs are calculated by a few number of steps which are the following ones:

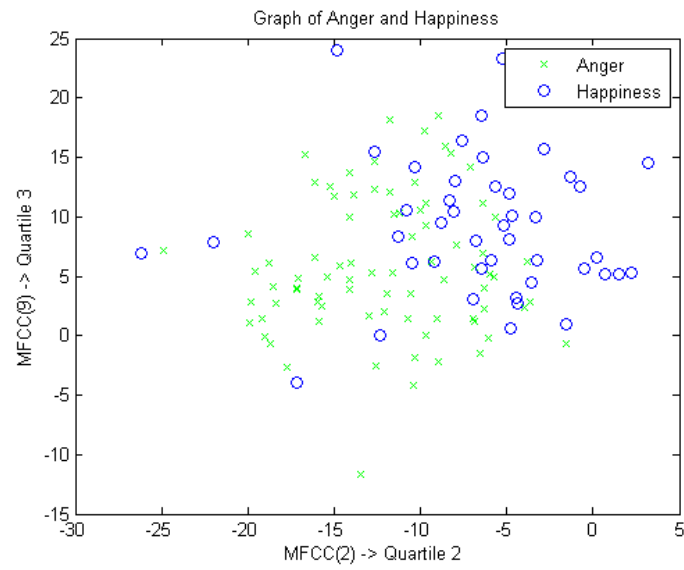
1. Frame the signal in small frames
2. Apply Discrete Fourier Transform
3. Take Log of the amplitude spectrum
4. Mel-scaling and smoothing
5. Compute Discrete Cosine Transform (DCT)
6. Keep the DCT coefficients 2-13

All of these steps have been done using the same library as before and openSMILE. Once the MFCCs have been calculated, the next step is to use them to obtain the features.

In this case, the features that have been obtained from the MFCCs are:

- Extremes: Maximum, Minimum and Range.
- Moments: Variance, Standard deviation, Skewness, Kurtosis and Mean.
- Percentiles: 1st, 2nd and 3rd quartile.

Now the features have incremented and they are more different between emotions which before were very similar with the features extracted. For example, Figure 10 shows how now, using the MFCCs related features, the Happiness and Anger can be differentiated and separated better. In this case the features used are the Quartile 2 of the MFCC(2) and the Quartile 3 of the MFCC(9).



**Figure 10: Graph of Happiness and Anger using two MFCC related features (Quartile 2 of the MFCC(2) and Quartile 3 of the MFCC(9)).**

### 3.3 Classification

The last step of this project is the classification. The classification is the problem of identifying, using a set of previous observations, the category of a new observation. When we talk about machine learning, there are two types of algorithms, the supervised and the unsupervised algorithms.

The supervised learning algorithms make predictions based on a set of examples. They are algorithms that use a labelled database, which means that every set of features belongs to a class [35]. In unsupervised learning, data points have no labels associated with them. Instead, the goal of an unsupervised learning algorithm is to group the data into clusters and find different ways of organizing the data [35].

In this project, the data obtained comes from examples where we know the emotions expressed and they can be labelled; therefore the project will use a supervised learning algorithm. Taking into account the classifiers explained in the last section, one of the bests and simplest ones to implement is the Naïve Bayes classifier. This algorithm is also more suitable for a compact hardware implementation in the future; hence, this classifier will be the one that is going to be used. The Naïve Bayes classifier is an algorithm which uses the Bayes theorem to classify. Bayes theorem is stated mathematically as the following equation:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \quad (1)$$

Where,

- $P(A)$  and  $P(B)$  are the probabilities of observing  $A$  and  $B$  without regard to each other.
- $P(A | B)$ , a conditional probability, which is the probability of observing event  $A$  given that  $B$  is true.
- $P(B | A)$  is the probability of observing event  $B$  given that  $A$  is true.

#### 3.3.1 How Naïve Bayes works

Naïve Bayes classifiers can handle an arbitrary number of independent variables whether continuous or categorical, continuous in the case of emotion classification based on the speech. Despite its simplicity, Naïve Bayes normally perform better than more sophisticated classification methods. The decision rule for this classifier uses the maximum a posteriori technique (MAP):

$$\underset{y}{\operatorname{argmax}} P(y|x_1, \dots, x_n) \quad (2)$$

where  $y$  is a class variable and  $x_1$  to  $x_n$  are dependent feature vectors.

The steps that this classifier follows to choose the class where a new sample belongs to are the following ones:

1. Divide the database of features into a training and a test set. In our case, 70% training and 30% test.
2. Find the probability  $P(y|x_1, \dots, x_n)$  that will be the probability that a new sample belongs to an emotion  $y$  given the input features  $x_1, \dots, x_n$  computed as:

$$P(y|x_1, \dots, x_n) = \frac{P(x_1, \dots, x_n|y)P(y)}{P(x_1, \dots, x_n)} \quad (3)$$

Firstly, computing the probability  $P(y)$  of each class, these probabilities can be calculated from the training database. As an example, if the number of happiness samples is 30 and the total of the training database is 150 samples, this is the probability:

$$P(y) = \frac{30}{150} = 0.2 \quad (4)$$

Secondly, according to the naïve independence assumption, the probability of generating instance  $x_1, \dots, x_n$  given class  $y$  can be changed to this one:

$$P(x_1, \dots, x_n|y) = \prod_{i=1}^n P(x_i|y) \quad (5)$$

The different Naïve Bayes classifiers differs mainly by the assumptions they use regarding the  $P(x_i|y)$  distribution. In our case, the assumption that it is going to be used is that the likelihood of the features is Gaussian as we can observe in the Equation 6.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \quad (6)$$

Hence, the training samples are again used to create a model which is going to be used to compute the probability  $P(x_i|y)$  of each feature of the new sample for every emotion. So the result will be a probability  $P(x_1, \dots, x_n|y)$  for each class  $y$ .

Once the numerator probabilities are obtained, the next one step is the for  $P(x_1, \dots, x_n)$ . It is a probability that is constant for each class  $y$ , therefore when the maximum argument is computed, it won't affect the decision, Equation 7.

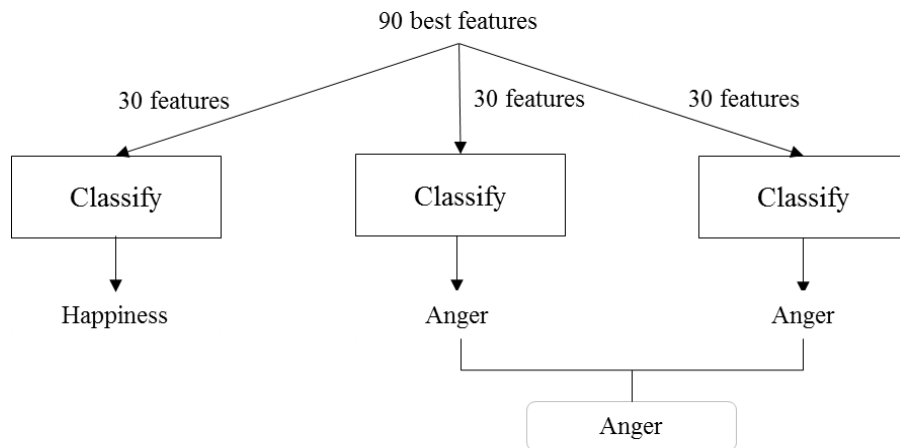
$$decision = \underset{y}{\operatorname{argmax}} P(y) \prod_{i=1}^n P(x_i|y) \quad (7)$$

Where decision is the emotion classified by the classifier using the set of features computed given a new sample. The decision will be only one, but the results of the probabilities  $P(y|x_1, \dots, x_n)$  will be each one of the probabilities of being a class or another, for example, the values obtained will be 30% anger, 20% happiness and 50% sadness. This means that we will know the probability by which a new sample might be belonging to all classes and it can be used to improve in many ways the emotion classification. One of the improvements could be weighting the emotions and using another classification method as emotion recognition by facial expressions to also weight them and obtain a decision based from both classifications.

### 3.3.2 Classification improvements

In all machine learning systems, the more we try different ways, the more probabilities exist to obtain better results. In this system, using the classifier implemented with the aim of classifying emotions we can develop an algorithm to obtain a ranking of all the features in order to use the best ones (without taking into account the correlation between features) to classify emotions. The algorithm consists of using the classifier model obtained from the training samples to classify the test samples, using only one feature by one ( $x_1$ ). The test samples are also labelled so it means that by the results obtained a ranking of each feature can be created; the feature with more accuracy on classifying will be the first one on the ranking, the second with more accuracy the second one, etc. Taking random features is way worse than use only a set of the best ones. Thanks to this step, the accuracy of the classifier has been increased about 10% which is a big improvement.

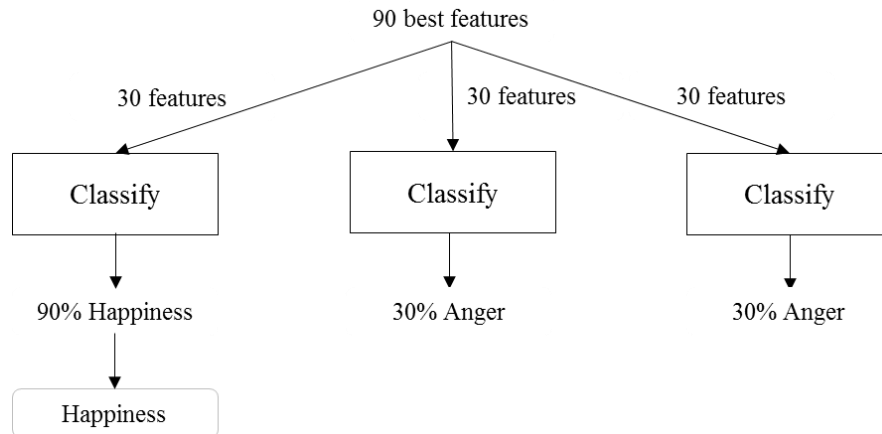
The next idea is about using the features obtained from the ranking to develop a layer-based algorithm. After knowing the best features based on the ranking created before, the algorithm will use a reduced number of features to classify a new sample. It will do that a few iterations until all the best features are used and after that it will classify the new sample as the emotion classified more times. The following figure shows how it works:



**Figure 11: Layer-based algorithm**

The result obtained by this layer-based method is not the best result because it can be improved by the probabilistic hypotheses of the Naïve Bayes classifier. It means that each time the classifier decides it computes the probability of each emotion, looking at the last example (Figure 11), after each classification only the emotion with more weight has been taken into account but the thing is that the classifier knows that the first classification as Happiness is maybe only the 40% and the other 60% is between the other emotions. Knowing this, when the classifier decides which one of the emotions is the most recognized, it will put weights onto them. An example following the last one could be:





**Figure 12: Weighted classification**

In this case, the results have been:

- First layer obtained 90% probability that the sample is Happiness
- Second and third layer obtained that the probability is 30% Anger in both of them.

Second and third layer obtained 30% probability of being Anger and 30% probability of being Happiness. The classifier will have to decide if what is more trustable, a classification of 90% of Happiness in one layer or a classification of 30% of Anger in two layers. It will be more trustable a layer that obtains a 90% probability that the sample is Happiness than two layers with 30% probability of being Anger.

With this last improvement, the accuracy of the results has been increased by approximately 5%, which helps to determine the emotion better.

## 4. Implementation and Results

The aim of this project was the development of an Emotion Recognition classifier using C programming language. C is a low level language which is useful because it can be crafted to run very fast and with very small memory footprint.

### 4.1 Feature extraction using OpenSMILE and CSV database

In order to develop all the code to implement the whole system, OpenSMILE library have been used to extract all the features. OpenSMILE is an open library written in C that has been created with the aim of helping to create systems based on emotion recognition from the speech. It has a lot of functions but the main one for this project has been the extract function to obtain all the features wanted from a WAV speech signal. To do that, OpenSMILE library uses configuration files that need to be changed in order to obtain the desired features. In this project, the necessary features are MFCCs and pitch related features; in the configuration file the next steps have been followed to extract the pitch:

1. Read a WAV input file

```
[waveIn:cWaveSource]
writer.dmLevel=wave
filename=\cm[inputfile(I){test.wav}:name of input file]
monoMixdown=1
```

2. Frame it into 25ms windows

```
[frame:cFramer]
reader.dmLevel=wave
writer.dmLevel=outp
frameSize = 0.025
frameStep = 0.010
frameCenterSpecial = center
```

3. Compute the pitch using an autocorrelation and cepstrum based method

4. Use the results of every frame to compute the statistical functions

```
// statistical functionals
[functL1:cFunctionals]
reader.dmLevel=smo
writer.dmLevel=func
copyInputName = 1
; frameMode = var will enable the functionals component to listen for messages from the turn detector
frameMode = full
functionalsEnabled=Extremes;Regression;Moments;Percentiles
Extremes.max = 1

Moments.variance = 1
Moments.stddev = 1
Moments.skewness = 1
Moments.kurtosis = 1
Moments.amean = 1
Percentiles.quartiles = 1
```

5. Use the output results to create a CSV file that contains all the extracted features from an audio file.

The CSV file obtained as an output has the format mentioned in the Feature extraction section but it is still not labelled. Each one of the WAV files from the database has a name with the emotion written in it, as an example, 03a01Fa.wav is an audio file that expresses Happiness (F=Freude in German) so the code will read the name of the file and will add a label to the end of the CSV file corresponding to the 6<sup>th</sup> character of the filename. It will be done for every one of the WAV files in the database and it will be saved as a database of features, the following pictures show how:

```
if [ $initial = "W" ]
then
sed -i 's/;/1/' selfconf.csv
cat selfconf.csv >> databasefull.csv
```

**Figure 13: Saving the CSV file labelled of an anger sample to the database of features (bash code)**

The CSV format database consists on a file with text lines, each line corresponds to an audio file and has numbers separated by commas. The numbers are the features obtained and every line ends with a number from 1 to 5 that correspond to the label of the emotion of the file, for example, Anger is the number 1.

```
1 4.824718e+02;1.998782e+04;1.413783e+02;...;2.496349e+01;1
2 4.596327e+02;2.677791e+03;5.174738e+01;...;2.943095e+01;5
```

**Figure 14: First two lines of the features database**

Once the features database is created, it is necessary to randomize the samples and separate the database in two parts, the training (70%) and the testing database (30%).

## 4.2 Classification (Training and Testing)

As it has been explained in previous sections of the thesis, the step after the extraction of the features is the classification. The classification has two parts, first the algorithm should create a model with the training samples and after that, use it as a classifier for the new samples.

The training step of the classifier, Naïve Bayes in this case, consists of using each and every features of the training database, to obtain the mean and the variance of every feature for every emotion. The next figures show the steps followed:

1. Read the training database and save it as an array to be able to do the calculations of the features' values

```
while(i<numrows){
    switch ((int)all[i][numcols]){
        case 1:
            //---anger---
            j=0;
            while(j<numcols){
                anger[a1][j]=all[i][j];
                j++;
            }
            for(x=0;x<numcols;x++)
            {
                meansvars[0][x]=meansvars[0][x]+anger[a1][x];
            }
            a1++;
            break;
        case 2:
            //---boredom---
```

Figure 15: Part of the code to extract the features from the training database and save them as an array

2. Compute the mean and the variance of each feature from each emotion and save them into different files. The lines represent the emotions and the numbers separated by spaces are the mean of each feature.

```
1 426.322571 15219.443359 121.887825 -0.081211 2.024829 :
2 381.781586 7640.037598 85.219223 1.157055 5.997751 88.4
3 394.874725 11171.798828 104.279732 0.344753 2.500221 1:
4 407.841797 12632.688477 110.638039 0.299169 2.713905 1:
5 421.889862 15443.558594 122.323570 0.027726 2.097480 1:
```

Figure 16: File with the means of the features of each emotion

As explained in Classification improvements, the features will be used one by one in the classifier to obtain the features with best results. They will be saved in a file and 100 of them are the ones that are going to be used in the testing step. The first column of the next figure shows the position of the feature in the ranking, the second column is the number of the feature and the third column represents the number of good classifications.

1	77	107
2	169	103
3	82	100

**Figure 17: Ranking of features (Top 3)**

After the training step and the ranking, the values obtained are used to compute the probabilities that will show the emotion of a new sample. The development made to obtain the classification begins firstly by obtaining the features of a new sample from the testing database. Once the features' values are extracted from the database, the classifier will use the model obtained from the training step to compute the probability with which the new sample may belong to each class (Figure 18), done by using this formula  $\prod_{i=1}^n P(x_i|y)$ . As explained in Classification improvements, the classifier will classify an emotion using a divider of 100 and then will use the different results of the layers and the weights to compute the final decision. This will be done for each one of the samples of the testing database and since we know the real emotions of the samples we can compare the emotion classified by the classifier and the real one to know the accuracy of the classifier (Figure 19).

```
for (n=0;n<numrows;n++){
    s=0;
    for (i=0;i<nfeatused;i++){

        Pdcjs[0] = Pdcjs[0]*((1/sqrt(2*3.14159265358979323846*angervariance[featuresa[i]]))*exp(-(pow((testDB[n][featuresa[i]]-angermean[featuresa[i]]),2)/(2*angervariance[featuresa[i]]))));
        Pdcjs[1] = Pdcjs[1]*((1/sqrt(2*3.14159265358979323846*sadnessvariance[featuresa[i]]))*exp(-(pow((testDB[n][featuresa[i]]-sadnessmean[featuresa[i]]),2)/(2*sadnessvariance[featuresa[i]]))));
        Pdcjs[2] = Pdcjs[2]*((1/sqrt(2*3.14159265358979323846*neutralvariance[featuresa[i]]))*exp(-(pow((testDB[n][featuresa[i]]-neutralmean[featuresa[i]]),2)/(2*neutralvariance[featuresa[i]]))));
        Pdcjs[3] = Pdcjs[3]*((1/sqrt(2*3.14159265358979323846*fearvariance[featuresa[i]]))*exp(-(pow((testDB[n][featuresa[i]]-fearmean[featuresa[i]]),2)/(2*fearvariance[featuresa[i]]))));
        Pdcjs[4] = Pdcjs[4]*((1/sqrt(2*3.14159265358979323846*happinessvariance[featuresa[i]]))*exp(-(pow((testDB[n][featuresa[i]]-happinessmean[featuresa[i]]),2)/(2*happinessvariance[featuresa[i]]))));
    }
}
```

**Figure 18: Part of the code for computation of probabilities**

```
if(k+1 == (int)testDB[n][numfeat]){
    numok[k]++;
}

percent=((float)numok[0] / (float)numtotal[0])*100;
printf("ANGER: %f\n",percent);
percent=((float)numok[1] / (float)numtotal[1])*100;
printf("SADNESS: %f\n",percent);
```

**Figure 19: Obtaining the accuracy of the classifier**

All the algorithms are compacted in one bash code that allows the user to use all the functionalities of the program. The next figures show how it works and the functions of the program. Figure 20 is the code of a function of the program; it compiles all the necessary files that are going to be used by another call. The calls to each one of the functionalities of the program can be shown in the terminal writing this help command:

**>>EmoRecognition.sh -h**

Figure 21 displays the output of the help command; all the functions are explained with detail to be able to work easily with the program.

```
shift # past argument
;;
-c|--compile)
compile=1
break
shift # past argument
;;
```

```
#compiling the program
if [[ $compile -eq 1 ]]
then
echo "Compiling the program ..."
gcc Ccodes/NaiveBayesMAP.c -o naive
gcc Ccodes/naivefeatureselection.c -o naivefeatureselection -lm
gcc Ccodes/NaiveBayesclasstestDBweighted.c -o naiveclasstestDB -lm 2>/dev/null
gcc Ccodes/NaiveBayesclasstartweighted.c -o naiveonetest -lm 2>/dev/null

echo "The program has been compiled succesfully."
fi
```

**Figure 20: Code that compiles all the files (option -c|--compile)**

```

Usage: EmoRecognition [-option (value)] ...

-h|--help
    Show this usage information

-d|--database <databasepath>
    Creates the features database using the <databasepath> where the .WAV files are.
    The completely database is called databasefull.csv and it is divided into
    the databasetrain.csv (70%) and the databasetest.csv (30%).
    Path to database document with the format:
    /home/user/Desktop/wav/03a01Fa.wav
    /home/user/Desktop/wav/03a01Nc.wav
    -
    -
    -

-c|--compile
    Compiles the full program to start using it

-T|--train
    Trains the Naive Bayes classifier using the databasetrain.csv and select the
    best features using Bayes

-t|--test <features> <layers>
    Tests the databasetest.csv and shows the probability of success for each emotion.
    You can select the number of features that you want to use (1-100) and the number
    of layers of the classifier

-e|--eliminate
    Deletes all the files generated by the other options.
    CAUTION! If eliminate is used, you will need to compile and follow the next steps
    from the beginning.

-s|--start <filepath>
    Start classifying the file and obtain which emotion has. Using 100 features and 10 features
    per layer as default.

-r|--record <name>
    Start recording and save it with the name choosen.

-p|--play <name>
    Start playing the <name>.WAV file choosen.

-o|--openconf
    Open the configuration file selfconf.conf and change the desired features to extract.
    The configuration file works as the configuration files from the openSMILE .
    CAUTION! don't try to change without knowledge or it won't work.
    "

```

Figure 21: Functions implemented in the program

### 4.3 Simulation Results

The Berlin Emotion Database contains utterances for 7 different emotions but this project focused only on 5 of them; Happiness, Anger, Fear, Sadness and Neutral. The aim of the project was to develop a software capable to recognize emotions from an audio file as an input and do it quickly to create in the future a live emotion recognition system, and after all, the project has accomplished its goals.

The results obtained were initially very bad, around a 50% of accuracy, because the features used were pitch and MFCCs related features but only a few of them such as maximum, minimum, mean and variance and without a selection of the best ones. After that, the use of the OpenSMILE library helped to extract more features with more exactitude. Using the features extracted from the OpenSMILE software and developing the layer-based method explained before, the results increased about a 20%, obtaining an accuracy of 70%.

Developing the weighting method explained before, the results have been around the 75% accurate, which is pretty similar to the results that other people have obtained with this kind of classifier and features [18,20-27]. However, the main difference between this project and the others is that the programming language used is C (low-level language) so it can be useful to develop a final software able to be used in a robot for different kind of purposes as mental health treatments.

The results of the confusion matrix shown in the Table 7 can be described as good but the main trouble found is the prediction of happiness that sometimes the classifier confuses the happiness with the sadness and it can be a problem. But in spite of it, the overall recognition success is 78.22%.

Classified as →	Anger	Sadness	Neutral	Fear	Happiness
Anger	<b>87</b>	0	0	4	9
Sadness	0	<b>100</b>	0	0	0
Neutral	0	7	<b>79</b>	7	7
Fear	16	16	4	<b>60</b>	4
Happiness	28	0	0	9	<b>63</b>

**Table 7: Confusion matrix expressed in % of the final results obtained**



Current work (NB)	NB	KNN	GMM	SVM
78%	77%	81%	68%	67%

**Table 8: Results of different classifiers**

The results of the different classifiers in some papers found, as said afterwards, are very similar to ours. The result that have the most importance is the result obtained using a Naïve Bayes classifier which is almost the same that the one obtained in this thesis, it depends always on the features and the data, but in this case, the database is the same and thanks to the weights, layers and the feature selection, in this thesis, the results have been a little bit more accurate.

## 5. Conclusions and Future Works

In this thesis the objective has been the development of a software capable of recognizing emotions through the speech with good precision and written in C programming language. The objective has been accomplished successfully, obtaining 78% of accuracy. The software is efficient and quick, written in a language that, afterwards, will be useful for machines/robots.

The three most important and highlighted steps have been searching and understanding the database, transform the audio files to features, creating CSV files and select the best ones for our classifier, and finally obtaining the results with the help of the Naïve Bayes classifier, using weights and layers properly. We can say that the most difficult step has been the development of the classifier because of the few available resources about emotion recognition in C language.

In conclusion, although the addition of some more features could lead to a better performance, they will also increase the process time which will not satisfy our needs. Moreover, a 78% of accuracy, thanks to a code written in C, using a quick classifier and obtaining from it probabilistic hypotheses, is an acceptable result.

### 5.1 Future work

For the current research, the features that have been extracted from the WAV files are MFCCs and pitch related features, therefore, for future work maybe a different set of features, including both used ones and new ones like rhythm, intensity, energy or LPCC could be an improvement to our system. Changing the classifier and trying a different one can be also a good improvement to obtain a better accuracy in the classification. To do that, a good idea is to use high-level programming languages with libraries to find out the best configurations of the system to obtain the best results, after that, if for example is necessary to have a low computational cost, it will need to be written in a low-level language but only once and with the best configuration. Low-level languages need a large code to do functions and by contrast, the high-level codes are less long. The high-level languages are easier and with more libraries for all the machine learning purposes, Python is one of the best languages with lots of libraries for this kind of projects. Different databases can also be tried, for a better recognition with not-acted samples, a not-acted database can be useful. Obtaining the best emotion recognition accuracy is the main aim of the future work, and with the options proposed better results can be obtained.

More work is needed to improve the system so that it can be used in real-time speech emotion recognition. Finally, a bigger improvement for the system could be the addition of different types of emotion recognition as the emotion recognition by facial expressions or by the hearth beating and take a decision based on the classification of each one of the different emotion recognition parts.

## 6. References

- [1] R. W. Picard, "Affective Computing," 1995.
- [2] M. M. E. A. M. S. K. and F. K. , "Survey on speech emotion recognition: Features, classification schemes, and databases," 2011.
- [3] S. G. Koolagudi and K. S. Rao, "International Journal of Speech Technology," Volume 15, 2012.
- [4] C.Vinola and K.Vimaladeví, "A Survey on Human Emotion Recognition Approaches, Databases and Applications," 2015.
- [5] Y. W and P. K, "A Study of Emotion Recognition and its Applications," 4th International Conference, MDAI, Springer, 2007.
- [6] "Speech emotion recognition," [Online]. Available: [www.emospeech.net](http://www.emospeech.net).
- [7] M. A. Tischler, C. Peter, M. Wimmer and J. Voskamp, "Application of emotion recognition methods in," 2007.
- [8] C. Stickel, M. Ebner, S. Steinbach-Nordmann, G. Searle and A. Holzinger, "Emotion Detection: Application of the Valence Arousal Space for Rapid Biological Usability Testing to enhance Universal Access," 2009.
- [9] R. Plutchik, in *Emotion: Theory, research, and experience: Vol. 1. Theories of emotion*, New York, Academic, 1980.
- [10] S. Handel, "Classification of Emotions," 2013.
- [11] D. A. Bartels, "Berlin Database of Emotional Speech," TU Berlin, [Online]. Available: <http://www.emodb.bilderbar.info/>.
- [12] "Expressive Synthetic Speech," [Online]. Available: <http://emosamples.syntheticspeech.de/>.

## *References*

- [13] "technische universität münchen," [Online]. Available: <https://mediatum.ub.tum.de/doc/1137841/780196.pdf>.
- [14] V. Hozjan, Improved Emotion Recognition with Large Set of Statistical Features, Geneva, Switzerland, 2003.
- [15] P. Oudeyer, The production and recognition emotions in speech: Features and algorithms, Sony CSL Paris, 2003.
- [16] Y. Pan, P. Shen and L. Shen, "Feature Extraction and Selection in Speech Emotion Recognition," Shanghai, China.
- [17] R. Kohavi and F. Provost, "Glossary of Terms: Special Issue on Applications of Machine Learning and the Knowledge Discovery Process," 1998.
- [18] W.-B. L. Chung-Hsien Wu, "Emotion Recognition of Affective Speech Based on Multiple Classifiers Using Acoustic-Prosodic Information and Semantic Labels (Extended Abstract)," Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, 2015.
- [19] K. E. a. K. L. D. Neiberg, Emotion Recognition in Spontaneous Speech Using GMMs, Pittsburg, 2006, pp. 809-812.
- [20] Y.-T. C. J.-H. Y. Y.-H. C. Tsang-Long Pao, "Emotion Recognition and Evaluation of Mandarin Speech Using Weighted D-KNN Classification," Department of Computer Science and Engineering, Tatung University, Taipei.
- [21] Z. C. X. C. Peipei Shen, "Automatic Speech Emotion Recognition Using Support Vector Machine," Shangai, China, 2011.
- [22] S. Prasomphan, "Improvement Of Speech Emotion Recognition with Neural Network Classifier by Using Speech Spectrogram," Department of Computer and Information Science, Faculty of Applied Science, King Mongkuts University of Technology North Bangkok, 10800, Thailand.
- [23] C. C. S. S. Chandrakala, "Combination of generative models and SVM based classifier for speech emotion recognition," Atlanta, Georgia, USA, 2009.
- [24] C.-N. A. Theodoros Iliou, "Comparison Of Different Classifiers for Emotion Recognition," Cultural Technology and Communication Department, University

## References

of the Aegean, Mytilene, Lesvos Island, 2009.

- [25] W.-Y. L. Y.-t. C. J.-H. Y. Y.-M. C. C. S. C. Tsang-Long Pao, "Comparison of Several Classifiers for Emotion Recognition from Noisy Mandarin Speech," Department of Computer Science and Engineering, Tatung University .
- [26] S. M. N. P. Raul B. Lanjewar, "Implementation and Comparison of Speech Emotion Recognition System using Gaussian Mixture Model and K-Nearest Neighbor techniques," Department of Electronics, Dr. Babasaheb Ambedkar College of Engineering and Research, Nagpur, Maharashtra, India, 2015.
- [27] H. G. Martin Gjoreski, "Machine Learning Approach for Emotion Recognition in Speech," Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia, 2014.
- [28] C. Prakash, V.B.Gaikwad, R. R. Singh and O. Prakash, "Analysis of Emotion Recognition System through Speech Signal Using KNN & GMM Classifier," Shri L.R Tiwari College Of Engineering, Mira Road, Mumbai Mumbai University.
- [29] P. D. W. Sendlmeier, "Berlin Database of Emotional Speech," 1999. [Online]. Available: <http://www.expressive-speech.net/>.
- [30] C. Bishop, in *Pattern recognition and machine learning*, Berlin: Springer, ISBN 0-387-31073-8, 2006.
- [31] K. Chan, J. Hao, T.-W. Lee and O.-W. Kwon, "Emotion Recognition by Speech Signals," University of California, San Diego, USA.
- [32] D. Gerhard, "Pitch extraction and fundamental frequency: History and current techniques," Regina, Saskatchewan, CANADA, 2003.
- [33] F. Eyben, F. Wening, M. Woellmer and B. Schuller, "The Munich Versatile and Fast Open-Source Audio Feature Extractor," [Online]. Available: <http://audeering.com/research/opensmile/>.
- [34] D. a. Mermelstein, "IEEEexplore," 1980. [Online]. Available: [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1163420&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D1163420](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1163420&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D1163420).

## *References*

- [35] M. Mohri, A. Rostamizadeh and A. Talwalkar, "Foundations of Machine Learning," The MIT Press , 2012.

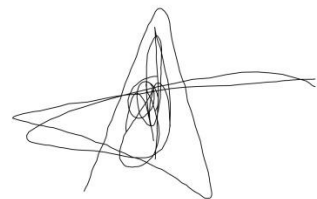
# Declaration

Hereby, I declare, this present work has been drawn up without inadmissible aid of third parties and without usage of other than mentioned resources. Further sources or indirectly appropriated data and concepts are identified by stating the source.

This work has not been presented to other examination procedures, neither nationally, nor in foreign countries, in the same or in a similar form.

Vienna, 30th July 2016

Emotion recognition by the speech, using a Naive Bayes classifier; Àngel Urbano Romeu

A handwritten signature in black ink, consisting of a series of loops and a long horizontal stroke extending to the right.