Exploring the Emotify dataset: genre and emotion prediction with musically meaningful MIR features

Sergi Andreu[1]
saandreu@kth.se

Carsten van de Kamp[1]
ctvdk@kth.se

[1]MSc. Computer Simulations for Science and Engineering
KTH Royal Institute of Technology

November 12, 2021

ABSTRACT

Music is a powerful and common tool for mood regulation, but it is hard to predict the induced emotion for a musical piece. We explore this problem with the Emotify dataset, which contains 400 songs from four different genres, annotated on the Geneva Emotional Music scale. Using support vector machines, k-nearest neighbour classifiers and convolutional neural networks, we make an attempt to predict genre and induced emotion, using musically meaningful features directly extracted from the waveform. Classification methods perform better than making a random choice, but having non-trivially separable data on the feature space prevents us from obtaining a high accuracy. For regression methods for emotion prediction it turns out to be hard to come up with performance measures. We conclude that the small dataset with high-dimensional samples and noisy labels leads to an ill-posed problem, for which it is not easy to find good features. A bigger annotated dataset would open up the possibility for more sophisticated approaches.

1. INTRODUCTION

For many people, music is an important part of their lives. It has even been shown that people listen to music for three main reasons: regulating arousal and mood, achieving self-awareness and as an expression of social relatedness, where the latter has been judged to be less important compared to the other two [1].

Using music for mood regulation is powerful and common [2], but it is crucial that the listener can select the music that has the desired emotional effect. With the enormous online music databases of these days, emotion-based music recommendation systems have become immensely popular and are among the most successful methods to help the listener explore these databases to select his music [3]. Applications range from personalised music in bars or restaurants to helping people to relieve themselves from stress and negative thoughts [4].

Predicting the induced emotion of a certain musical piece is however far from straightforward. A main reason for this is the lack of large annotated datasets for supervised learning [5]. These datasets are also hard to generate. Since the annotations are subjective by nature and also depend on the current mood of the person being asked and one would need a lot of annotations to get a good impression of the actual emotion that is induced by a musical piece. Moreover, existing models often lack interpretability or trade performance for explainability [6].

In this work we want to explore this matter by considering the Emotify dataset [3]. In an online Facebook game, participants could rate 400 songs from four different genres on the GEMS scale (Geneva Emotional Music Scales) [7]. Using various methods method from the field of machine learning we try to build a predictive algorithm for genre and emotion based on musically meaningful features extracted from the songs in this dataset. The goal is to gain insight into what musical features vary for different musical genres and what features contribute to induced emotion. We focus on explainability of our choices and interpretability of the results, rather than a state-of-the-art performance score.

Four research questions motivating this work are:

(*Q1):* What features contribute to the classification of different genres?

(*Q2):* What features contribute to different induced emotions and how do these differ from the genre classification contributing features?

(*Q3):* Is it possible to interpret the relevant features for these problems in a meaningful way?

(*Q4):* Is transfer learning possible and reliable for music genre and induced emotion prediction, and do these approaches generalize outside of the considered dataset?

Moreover, it is of particular interest to what extent these questions can be answered with the 'small' dataset under consideration.

This text will give an overview of our attempts to find patterns between features from Music Information Retrieval (MIR) and the labels genre and perceived emotion, resulting from the Facebook game. Moreover, we try to show the

reader why the subject of music emotion recognition is so challenging. In chapter two we will familiarize the reader with the Emotify dataset and the various approaches that we undertook for genre and emotion prediction. We will not give an in-depth description of all machine learning methods, but we will motivate why we tried certain methods. Subsequently, in chapter three, we present and discuss the outcomes of the various analyses. Lastly, we draw conclusions in combination with recommendations for further research in chapter four.

This work constitutes the final report of the project in the course Music Informatics, taught at the KTH Royal Institute of Technology in Stockholm.

2. METHODOLOGY

2.1 The Emotify dataset

The Emotify dataset [3] is generated using an online Facebook game. Participants could select at most three emotions from the GEMS scale [7] that they felt strongly when listening to a song. This scale was proposed in 2008 to measure musically induced emotions specifically, and consists of nine emotional labels (amazement, solemnity, tenderness, nostalgia, calmness, power, joyful activation, tension, sadness). In total, the dataset contains 8404 emotional annotations of 400 one-minute song fragments in four genres (classical, electronic, pop, rock). We remark that every song excerpt has a different number of annotations as participants could skip songs. There are four (classical) songs that have a duration less than 60 seconds.

In order to convert the binary annotations to labels, we take the average number of times an emotion was reported for a certain song; the label 0 corresponds to the song never being reported to induce a specific emotion, whereas songs with emotions with the label 1 have been reported by all listeners. The emotion annotation data is observed to be sparse, and these averages are generally quite low, making it hard to turn the problem into a classification task. A natural threshold of 0.5 would make almost all songs 'emotionless' and an arbitrary threshold would make the predictor highly sensitive to the choice of this arbitrary threshold.

Hence we look at classification methods for genre prediction and mainly at regression methods for emotion prediction.

To the best of our knowledge, there has not been a similar research before using the Emotify dataset. The actual dataset has however been analyzed for i.a. finding patterns between genre and induced emotion and the role of native language, age and gender of the listener on the induced emotion [3].

The size of the dataset may not allow for using architectures or approaches shown successful in similar problems, but with different datasets. This, and our focus on *understanding* the difficulties of music emotion and music genre recognition, makes our approaches and assumptions based on the results we observe, rather than on previous work. In this sense, this work is not the continuation of specific previous work, but it can hardly be considered to be completely unique.

2.2 Feature extraction

As mentioned in the introduction, we want to extract musical features for which we can reason that they contribute to the distinction of genre or induced emotion, rather than feeding the waveforms or spectograms into a huge neural network and 'hope' that it performs well. For both genre classification methods and emotion regression methods we consider three different feature inputs: (a) a combination of 'simple features', (b) mel-frequency cepstral coefficients and (c) the output of a pre-trained convolutional neural network for music audio tagging. Next, we consider the three different options shortly and motivate why these are musically meaningful:

(a) Our 'simple features' input contains data extracted from the waveform in the form of zero crossings, spectral centroid, spectral variance and estimated static tempo. Here it would be particularly interesting to see to what extent these relatively basic and easily computable features contribute to distinct genres and induced emotions. These features are musically meaningful because intuitively, spectral centroid in combination with spectral variance is a measure of how the intensity over different frequencies is distributed over time and the static tempo gives an estimate of the musical piece is slow or fast. Zero crossings are very related to the actual waveform of the signal, where a high number of zero crossings reflect a noisy, or high frequency signal, with for instance a lot of percussive elements and a low number of zero crossings would reflect a smoother signal with a low fundamental frequency. In this report, zero crossings, spectral centroid and spectral variance are all computed with a frame size of 200 milliseconds and a hop size of 100 milliseconds (also to ensure feature vectors of equal length). However, we tried various parameters when experimenting. The static tempo estimate (BPM) is computed with librosa 0.8.1 [8].

(b) The mel-frequency cepstral coefficients (MFCCs) can be seen as a representation of the short term power spectrum of a musical piece. In particular, it is the cosine transform of the power spectrum on the nonlinear mel scale. It captures the timbral structure of a signal [9] and therefore they are potentially very helpful in genre and emotion prediction. In this work we compute the first 20 MFCCs with librosa 0.8.1, a window length of 250 milliseconds, a hop size of 100 milliseconds and an FFT size of 8192 samples.

(c) Lastly, we consider the output of the pre-trained musical audio tagger musicnn [10]. The amount of samples in the Emotify dataset prohibits learning relevant high-level features by feeding a neural network directly with the audio samples or its spectrograms. We hope that pre-computed models trained on similar tasks would learn relevant fea-

tures for our problem, as transfer learning has been observed to be useful in MIR tasks [11]. We use the *MTT vgg* model trained on the *MagnaTagATune* dataset. The predicted tags by this model include genre (classical, electronic, pop, rocks) as well as other high-level descriptors (male/female voice, synth, woman, piano, ...). We use some of the intermediate outputs of the pre-trained network as features, *pool5*, which contains 2×128 features. The *taggram* feautures are the outputs of *MTT vgg* (predicted tags), with a dimension of 1×50. The inputs used in our model are bigger than the ones used in *MTT vgg*, thus the dimensions of the features we use are $(2h, 128)$ and $(h, 50)$ for pool5 and taggram respectively, depending on an integer parameter h indicating the time of the slices used as inputs of our model.

It is of particular interest to see whether we can map the musicnn audio tags to the genres and emotions in the Emotify dataset.

The source code used for feature extraction and the various models is available at `github.com/sergi-andreu/ProjectMusicInformatics/` such that the results can be reproduced.

2.3 Genre prediction

In this section we will consider the various machine learning algorithms that we have used for genre classification, based on the aforementioned features. We will not go into deep detail regarding the methods, but we will comment on why we chose which methods and what we expect to see.

Since we intend to interpret the results, our first attempts will consider the simple features. We train three different classifiers on the simple features: **(i)** support vector machine, **(ii)** k-nearest neighbours and **(iii)** a convolutional neural network.

(i) The support vector machine (SVM) is a deterministic classification method that maximizes the width of the decision boundary in the four dimensional simple features space. To do multi-class classification, we use a 'one-versus-one' approach and we consider both linear SVM and radial basis function (RBF) kernel SVM for a non-linear decision boundary, with a regularization parameter $C = 1$. The SVMs are implemented using scikit-learn [12]. For each song we have three arrays containing data of zero crossings, spectral centroid and spectral variance for each window. Moreover we have a static tempo estimation. To reduce these to a four-dimensional feature space, we consider two options. Firstly, we take the median of the zero crossings, spectral centroid and spectral variance arrays, because we expect that the median might be a good discriminator between different genres. Secondly, we also consider a majority vote approach where we perform a classification on each window, and then take the final classification decision based on the class that has been predicted the most. In order to obtain reliable results, that will generalize over an unknown, but similar dataset, we perform 20 instances of 10-fold cross validation over different shuffles of the data. We will report the achieved average accuracy and standard deviation over the different shuffels, in addition to an averaged confusion matrix.

(ii) The k nearest neighbour (kNN) classifier determines the class of a datapoint in feature space by looking at the dominant class of the $k \in \mathbb{N}$ nearest neighbours of the datapoint in Euclidean sense. If good generalisation results are observed using kNN, the notion of *proximity* in an euclidean sense in the feature space would be relevant, and so the notion of clustering in that feature space. It is however a bad method to use for high-dimensional feature spaces, since the notion of proximity becomes irrelevant when there are not enough samples, thus we consider the median of the feature vector, as well as a majority-vote approach, and train the classifier using scikit-learn [12].

(iii) Convolutional neural networks (CNNs) are a class of artificial neural networks which modulate translation invariance, which is of interest in temporal feature arrays, as it is our case. They allow for using the temporal feature arrays without having to take the mean / median over time. They are universal approximators [13], so one should expect overfitting to the training data, since they can learn arbitrarily complex representations. The design choices for a CNN are not trivial, and have not been fully explored in any of the cases in this work. Our approach is based on choosing architectures and regularizations that avoid overfitting, considering the size of our dataset. These choices include:

- Slicing (time-wise) each feature array arbitrarily in training time, discarding the slices containing padded values (some songs have less duration than others). The duration used is usually 5-10s.
- Use 0.4-0.5 dropout rate in the dense layers.
- Choose a small number of neurons and a small size of the filter for convolutions.

We use Keras [14] and Tensorflow [15] for implementing the CNN. We do not use tempo here, since this would imply a constant temporal array that may hurt the learning process of the CNN.

A second CNN **(iv)** will be trained on the MFCCs features. MFCCs adds a higher-dimensional feature space which is believed to contain pitch information [9]. A higher-dimensional input space may admit a model with higher validation accuracies, but at the expense of overfitting if the number of samples does not scale with the number of unkown parameters. We made the same choices to avoid overfitting as in case (iii).

Lastly, we consider the musicnn audio tags as features in a **(v)** CNN model. The dimensionality of the input space is smaller, because the musicnn network has already been trained on a similar task. It is equivalent to consider using musicnn features as inputs as considering a bigger network with a high dimensional space (STFTs) with fixed

pre-trained weights on the first layers.

We note that some other attempts were discarded. As an example, CNN trained on estimated beat times of the audio samples, getting sparse binary arrays of the *clicks*. The sparsity of the arrays and avoiding long-duration arrays, makes training difficult. The influence of adding beat times to the other used features in the different approaches was not investigated.

The genre recognition models are evaluated mostly on the accuracy metric. This is considered to be a relevant metric in this setting since the data is balanced with respect to genre. Other metrics such as multi-class, F1 score, recall and precision are not reported, and we refer to the confusion matrix for full information on the performance.

2.4 Emotion prediction

This section we will contain an overview of the various machine learning algorithms that we have used for emotion prediction, based on the aforementioned features. As mentioned before, the emotion labels might be less suitable for classification and hence we will also consider methods for regression. Again, we will not go into deep detail regarding the methods, but we will comment on why we chose which methods and what we expect to see.

Our first attempts will consider the simple features. We look at three different approaches on the simple features: **(vi)** SVMs, **(vii)** ν-support vector regression and **(viii)** a convolutional neural network.

(vi) SVMs are a simple and robust method to see whether the classification task is separable. We consider linear and nonlinear RBF kernel SVMs, with a regularisation parameter $C = 1$. All nine emotions are considered separately, so this is a binary classification task. The emotion labels constitute averaged binary annotations thresholded at 0.5.

(vii) SVMs can also be used for regression tasks. They find an appropriate hyperplane to fit the data, where the parameter ν specifies how much error can be tolerated. In this work we train the regressor with scikit-learn [12] with $\nu = 0.5$, a penalty parameter $C = 1$ and a RBF kernel.

(viii) Compared to genre classification, here the CNN is used for regression. This comes at the cost of not having well-behaved and interpretable performance metrics for this problem, as well as complicating the decision on the loss function. Not being able to evaluate our model properly also makes architecture and hyperparameter selection difficult tasks. The chosen evaluation metrics are:

- Mean Squared Error (MSE): widely used as a loss function and as a validation metric. It however does not contain information on the direction of the emotion prediction, nor about the distribution of the error across samples. Since the emotion labels contain noise due to the subjective annotations, it may not be a reliable metric for this specific case of regression.
- Cosine Similarity (CS): gives information on the error of the direction in the predicted emotion. It is arguably a better metric to use for clustering musical samples. It however does not contain information on the *modulus* of the emotions.
- Coefficient of determination (R2): bounded by 1 and thus helpful to compare results. However, for high-dimensional cases, and using non-linear regression, the interpretation becomes unclear, especially for neural networks.

Other regression metrics are shown to be useful in our case, and they are included in the repository. However, we use as a loss function a combination of MSE and CS, and as validation measure we use the plots of the cumulative distribution function of both the squared error and the cosine error ($= 1-$ CS).

For emotion regression, we also attempt to use the MFCC features **(ix)** and the musicnn audio tags **(x)** as input for a CNN.

3. RESULTS AND DISCUSSION

3.1 Genre prediction

As a first step, we considered the simple features for genre prediction. Figure 1 shows a pair plot containing density function estimates of all four features and projected two-dimensional scatter plots for all combinations of two features. On these two-dimensional projection spaces the features for different genres are not separable, but since the features are contained in a four-dimensional space, this does not mean that they are not separable at all. To check this we train a linear and a nonlinear radial basis function (RBF) kernel support vector machine **(i)** on the dataset to find a decision boundary. The resulting confusion matrices are given in figure 2. From these we conclude that the data is also non-separable with linear or RBF kernels in its four-dimensional feature space. For 20 different shuffles of the dataset we trained the SVC using 10-fold cross validation. The achieved average accuracy is $40.9 \pm 0.5\%$ for the linear and $47.8 \pm 0.5\%$ for the nonlinear RBF kernel respectively, where the deviation is computed over the shuffles. It is worthwhile to note that the classifier achieves a better performance in distinguishing classical music compared to other genres.

We also attempted to train a SVM to classify the genre based on the features in each window, and then determine the actual genre of the song by performing a majority vote. Unfortunately the amount of data points becomes too large to train an SVM, which was done in Google Colab. Training was terminated after approximately 12 hours.

(ii) The kNN classifier is observed to have a poor performance compared to other approaches: using an optimal $k = 10$, we obtain $35.3 \pm 2.6\%$ accuracy. The notion of euclidean proximity in the space of simple features seems less relevant than other approaches for this problem. kNN

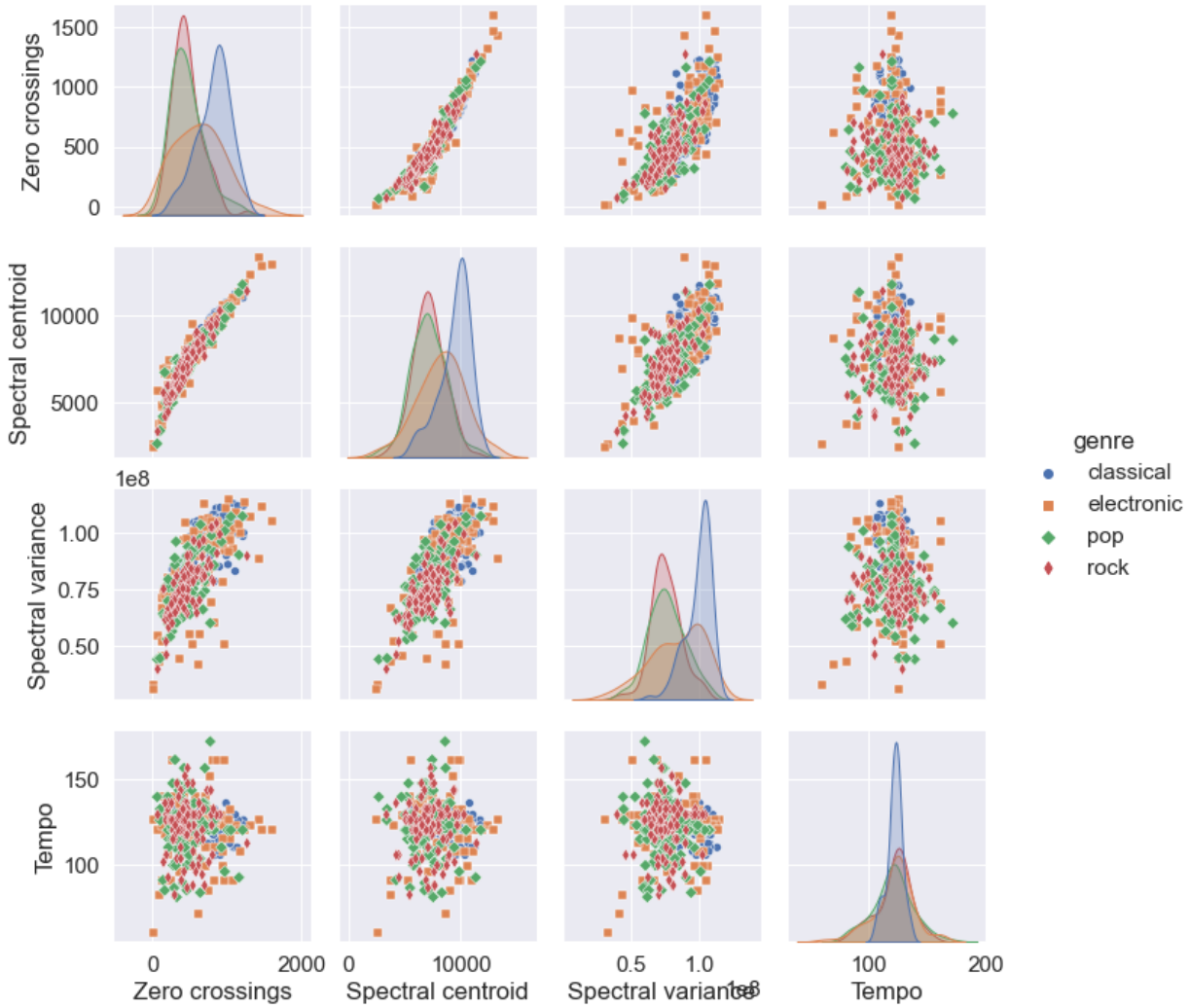
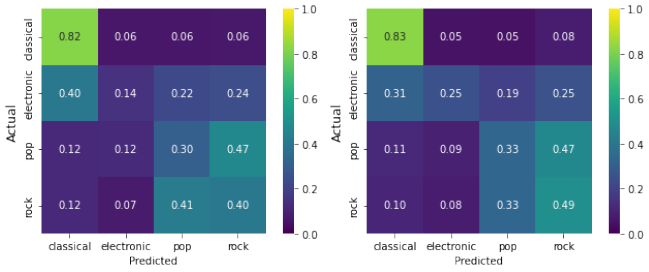


Figure 1: Pair plot of the simple features (zero crossings, spectral centroid (Hz), spectral variance (Hz), computed using 200 ms windows and 100 ms hop size, and estimated tempo (BPM) for the four different genres in the Emotify dataset. The diagonal plots contain density function estimates and the off-diagonal plots contain two-dimensional projected scatter plots of the different combinations of simple features.



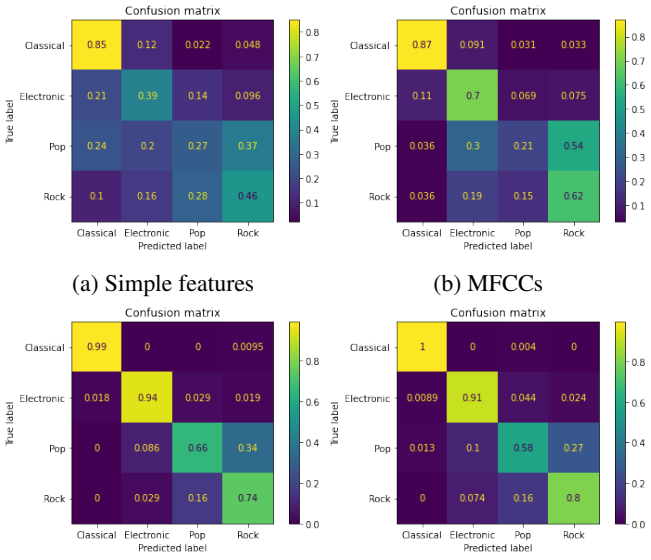
(a) Linear SVM (b) RBF kernel SVM

Figure 2: Confusion matrices for genre classification using SVMs based on the simple features. The confusion matrices are normalized over 20 different shuffle instances of 10-fold cross validation.

is discarded for more complex features since in higher dimensional feature spaces one would need more and more samples for distances to remain significant.

Using the array of simple features, with the extracted features at consecutive time windows, arguably takes into account more information that just looking at the components independently. The feature space could be extended such that it contains subarrays of simple features (for example the 3 selected simple features at n consecutive times). However, training that with simpler models (kNN, SVM, ...) is increasingly more difficult, and one does not emulate the translation invariance that we expect for this problem.

(iii) Using CNN was postulated as a better approach. However, the results are not conclusive: an accuracy of $47.8 \pm 0.5\%$ was obtained using the components separately, and one of $50 \pm 2\%$ is obtained when using CNN trained on the arrays of simple features, as opposed to element-wise approaches on SVM.



(a) Simple features (b) MFCCs

(c) Pool5 musicnn (d) Taggram musicnn

Figure 3: Confusion matrices for different features trained on CNN models for genre classification, using 5-fold cross validation with random weights initialization, 100 epochs and a batch size of 50

Although it is clear that the CNN *includes* the case of element-wise approach (neural networks can be seen as a generalization of a linear classifier, and convolutions are universal approximators), the random initialization of the networks, the fact that one has to select an architecture and hyperparameters, and the problem of how to efficiently and appropiately slice in time the simple arrays make the implementation of this methods a difficult task.

As observed in 1, using CNN, the accuracy results for both pool5 and taggram **(v)** are better than the ones obtained using SF or MFCCs **(iv)**, being $83 \pm 2\%$ and $82 \pm 2\%$ respectively.

The musicnn features are extracted from a model predicting tags, including our four genres. Therefore this does not come as a surprise, and it shows that transfer learning can be a good practice when having few amounts of data.

We restricted the architecture selection in SF and MFCCs to simple models in order to avoid overfitting, thus we reduce the variance at the cost of probably increasing the bias of our models. In this case, it seems reasonable to argue that the models trained on MFCCs and simple features learn some good representation of the features related to the musical genres, since the accuracy scores are $50 \pm 2\%$ and $60 \pm 6\%$ for SF and MFCCs respectively, thus better than random (25%).

In all cases, classical music obtained the higher accuracies, and pop and rock music seem to be confused by the models. Musical genres refer to a shared tradition or conventions in music, that can be arguably more pronounced in classical music, whereas in pop and rock music can be subjectively similar. The actual selection of musical pieces in the dataset has not been completely explored (that is, listened to) and it could give valuable information of which problem are we actually trying to solve.

Table 1 summarizes the results obtained from the attempted genre classfiers.

Table 1: Summary of the attempted genre classifiers and their performance.

	features	**method**	**accuracy**
		random genre	*25%*
(i)	simple	SVM - linear	$40.9 \pm 0.5\%$
(i)	simple	SVM - RBF kernel	$47.8 \pm 0.5\%$
(i)	simple	SVM majority vote	*too expensive*
(ii)	simple	kNN	$35 \pm 3\%$
(iii)	simple	CNN	$50 \pm 2\%$
(iv)	MFCCs	CNN	$60 \pm 6\%$
(v)	pool5	CNN	$\mathbf{83 \pm 2}\%$
(v)	taggram	CNN	$82 \pm 2\%$

3.2 Emotion prediction

Having considered the various approaches and results for genre prediction, which is a clear classification task, we will now focus on the results that we obtained in the various attempts for emotion prediction.

As mentioned in the methodology section, it is not straightforward to perform classification on the emotion labels. But before we consider the regression approaches, we will summarize the achieved accuracies for binary classification for each emotion, based on an arbitrary threshold of 0.5 with the simple features and a nonlinear RBF kernel support vector classifier in table 2 **(vi)**. The confusion tables are visualized in figure 4. We see that the classifier performs pretty bad (a random choice would achieve 50% accuracy in binary classification). Note that the classifier achieves a very high accuracy on classifying induced amazement. This is because the classifier labels all songs as not-amazement inducing in combination with the low number of amazement inducing songs. We observed that a linear SVM classifier performs even worse.

When transforming the emotion recognition problem into a classification problem, we have to decide a threshold for the features. A common choice of 0.5 makes the dataset unbalanced. Any arbitrary choice of this threshold value would get different results and performances, but in this case there is no motivation to why the threshold is chosen the way it is. Accuracy is not a good metric in this case, as opposed to in the case of genre recognition.

Treating the problem as a regression is arguably a better approach. However, it turns out to be difficult to find appropriate regression metrics for this problem. It is expected that in more specific applications there may exist a better way to annotate to labels, depending on the problem, but

Table 2: Accuracies achieved for binary emotion classification using RBF kernel SVMs based on the simple features. The accuracies are averaged over 5 differently shuffeled instances of 10-fold cross validation. Emotions with an average annotation greater than 0.5 are considered as induced by the corresponding song.

emotion	**accuracy** (%)
amazement	90.4 ± 0.2
solemnity	64.3 ± 0.9
tenderness	55.8 ± 0.7
nostalgia	$47,1 \pm 0.7$
calmness	43.4 ± 0.5
power	$73,9 \pm 0.3$
joyful activation	61.2 ± 1.0
tension	66.5 ± 1.4
sadness	53.4 ± 1.1

in this case it is not straightforward. For evaluation of the results here we argue that studying the distribution of both the squared error and the cosine error, is a good approach for keeping the formulation of the metrics general. However, this obstructs us to optimize the models efficiently, since the metrics used to compare these models are difficult to interpret. All of the metrics used (MSE, CS and R2) have drawbacks, as explained in the methodology section. This makes the interpretation of the results difficult.

The CNN **(viii - x)** results for the different features are summarized in table 3, the R2 metric is observed to be negative for all approaches, and CS and MSE are similar in all cases. Using the cumulative distribution functions of those can give more information of the results across different emotions, and also understand more about the error distribution in these.

Table 3: Validation metrics for different CNN models for emotion prediction, each using different features, obtained using 5-fold cross validation with random initialization, 100 epochs and a batch size of 50. SF refers to simple features.

	MSE $(\cdot 10^{-2})$	CS	R2
SF	3.77 ± 0.08	0.782 ± 0.004	-0.038 ± 0.004
MFCCs	3.71 ± 0.06	0.786 ± 0.001	-0.212 ± 0.007
pool5	4.05 ± 0.05	0.774 ± 0.003	-0.31 ± 0.03
taggram	$\mathbf{3.54 \pm 0.08}$	$\mathbf{0.784 \pm 0.006}$	$\mathbf{-0.114 \pm 0.008}$

The cumulative distribution of the errors is observed in figure 5. For all the selected features and models, the cumulative distribution function of the cosine error ($= 1-$CS) resembles that of a gaussian, and it is centered at $0.2 - 0.3$. This could indicate that the predictors are equally valid for determining the direction of the error in the predicted emotions. Regarding the distribution of the squared error per emotions, the shape of the curves for the MFCCs case is similar, and also the range of the errors. It is a difficult task to draw conclusions on the different performances of these approaches. However, the distribution of the errors of both

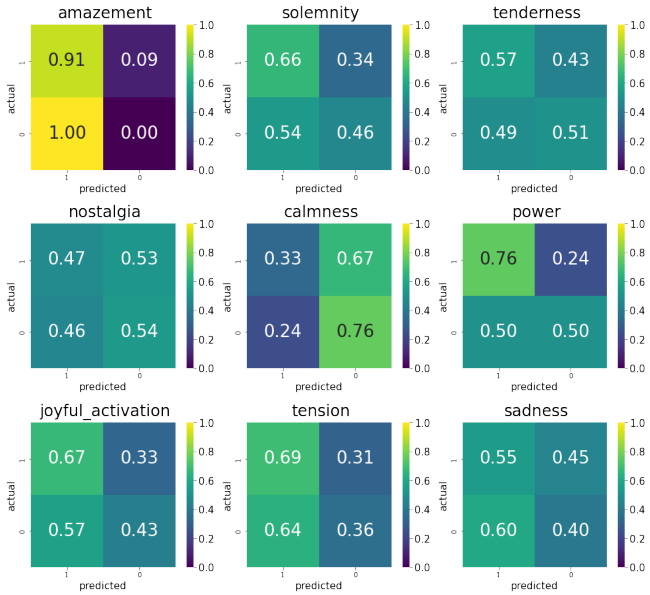


Figure 4: Confusion matrices for emotion classification using RBF kernel SVMs based on the simple features. The confusion matrices are normalized over 5 different shuffle instances of 10-fold cross validation. Emotions with an average annotation greater than 0.5 are considered as induced by the corresponding song and correspond to label 1 in the confusion matrices. For accuracy performance is summarized in table 2.

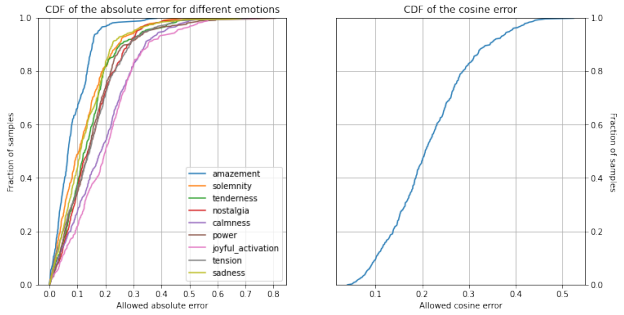
the square error and cosine error contain more information than a simple mean of these, although that information may be difficult to interpret.

A better evaluation of the performances would include the error as compared to the true value of the score. The relative error is here relevant since the distribution of the scores may have a big impact in the regression problem: for example, very few songs are observed to have an "amazement" score higher than 0.5. Therefore the error is not expected to be greater than 0.5 even if the regression predicts random scores from [0,0.5]. Therefore it is not trivial to deduce which emotions are easier to classify through only looking at the cumulative distribution function of the errors, without taking into account the distribution of those scores.

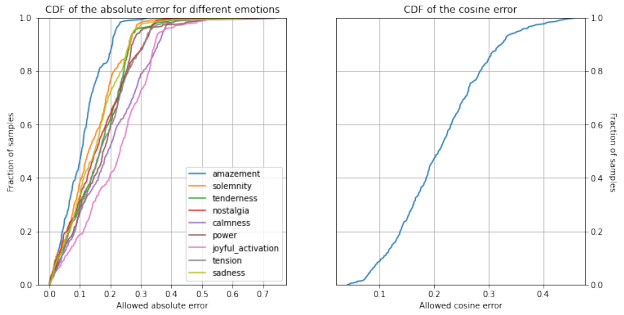
A more detailed analysis of this could be performed, but didn't fit in the time schedule for this work.

The relative error is arguably a bad metric, since some true scores can be 0. Moreover, one should penalise more the fact of not correctly predicting high scores. Finding reliable and robust metrics based on our intentions is needed to evaluate the performance of our models.

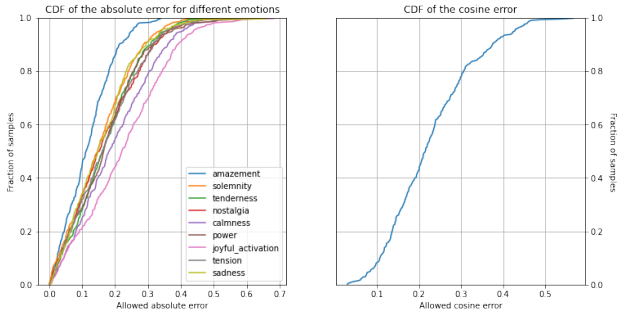
The MFCCs approach shows better performance metrics than the other approaches. Considering that the approaches also rely on the architecture and hyperparameter selection, there is noise in randomly slicing the arrays for validation



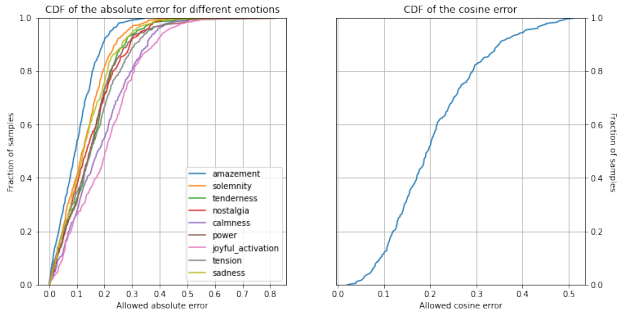
(a) Simple features



(b) MFCCs



(c) Pool5 musicnn features



(d) Taggram musicnn features

Figure 5: Cumulative distribution function of the squared error (left) and 1-cosine similarity (right) for the different CNN models for emotion prediction, using 5-fold cross validation with random initialization, 100 epochs and a batch size of 50.

(as opposed to majority vote or using all possible samples for validation) make the results inconclusive. However, a better performance on the MFCCs case could be explained by the fact that the simple features do not have enough information for the problem, and the musicnn network was trained on a problem not directly applicable to emotion recognition.

The gap between MFCCs and musicnn observed in genre recognition is not observed in emotion recognition. This could be a result of having an incorrect experiment setup, of having all models learning inefficiently or irrelevant features, or because of not having a strong correlation between high-level musicnn features and emotion labels.

Lastly, we mention that we also tried to train the ν-support vector machine for regression **(vii)** on the emotion prediction problem. It turned out to be too expensive to train, which was attempted in Google Colab. Training was terminated after 15 hours.

4. CONCLUSION

Predicting the genre and the induced emotion of a song using features directly extracted from the waveform is a challenging task. In this work we tried to explore this problem using the Emotify dataset.

Genre prediction is a clear classification task, but it turns out to be hard to find features that are easily separable in the feature space. A combination of zero crossings, spectral centroid, spectral variance and estimated tempo were used to train various classifiers, ranging from support vector machines and k-nearest neighbours to a convolutional neural network. The highest accuracy of about 50% was achieved using the convolutional network. We showed that higher accuracies can be achieved by training a convolutional neural network on more complex, higher-dimensional features. The best performance is achieved by training the classifier on the output of the pre-trained music audio tagger musicnn.

Regarding (*Q1*), the chosen simple features do not seem to be enough to achieve high accuracies on the predictions. More complex features extracted from bigger datasets are the ones given the best results. Transfer learning has been shown to be a good approach when dealing with small datasets for genre recognition (*Q4*). However, those features are obtained using convolutional neural networks, and they are difficult to interpret (*Q3*). Simpler models could be used on those features to assess correlations with the given labels, rather than relying on neural networks with random initialization, which difficult the choice of statistically relevant measures.

Emotion prediction can be seen as a classification task, but this introduces an arbitrary threshold to classify the data, which is undesirable. Training a radial basis function kernel support vector machine to this problem yields comparable results to making a random choice. Avoiding the thresholding problem comes at the cost of treating the problem as a regression task. This introduces new difficulties in defining a performance metric, and also an appropiate loss function. This also makes it hard to compare the

various approaches and evaluate our results.

It is expected that in more specific applications some metrics are natural to use. Since the proposal of our problem is rather vague, no decision has been made on *how* to optimize our models with respect to some reward.

Finding robust, reliable and interpretable metrics for regression task on emotion annotated musical pieces is difficult, and even more for such a general case. MSE cannot be considered to be a good indicator of the performance since the labels of the emotions contain non-trivial noise. In datasets with more annotations these difficulties could be overcome.

Both the approach and the results change from genre recognition to emotion recognition (*Q2*). The labels for the genre are already given, whereas for emotion we need to preprocess them. Some information regarding the characteristics of the annotators is discarded, and there is not an unique and reliable way to treat emotional annotations. The learned features for genre recognition are not completely transferable to emotion recognition, as observed by the musicnn performances on this task. It is again difficult to evaluate the performance of the models in this regression setting, thus we regard the results as being inconclusive.

All in all, the relatively small Emotify dataset with high-dimensional samples and subjective, and hence noisy labels leads to an ill-posed problem. It is hard to find good features, that ideally also allow for interpretability. Moreover, the training of more complex models on high dimensional features takes a lot of time.

Further research could focus on finding a bigger dataset containing high-quality annotations of induced emotion. This would open up the possibility for more sophisticated machine learning approaches. It could be worthwhile to invest more time in finding better features for the problem, in particular features containing beat or pitch information. And lastly, the regression problem should be better defined, such that there is a clear performance metric. Then it would also be possible to compare different approaches.

5. REFERENCES

[1] T. Schäfer, P. Sedlmeier, C. Städtler, and D. Huron, "The psychological functions of music listening," *Frontiers in Psychology*, vol. 4, p. 511, 2013.

[2] A. van Goethem and J. Sloboda, "The functions of music for affect regulation," *Musicae Scientiae*, vol. 15, no. 2, pp. 208–228, 2011.

[3] A. Aljanaki, F. Wiering, and R. C. Veltkamp, "Studying emotion induced by music through a crowdsourcing game," *Information Processing & Management*, vol. 52, no. 1, pp. 115–128, 2016.

[4] E. Kuntsche, L. L. Mével, and I. Berson, "Development of the four-dimensional motives for listening to music questionnaire (mlmq) and associations with health and social issues among adolescents," *Psychology of Music*, vol. 44, no. 2, pp. 219–233, 2016.

[5] J. de Berardinis, A. Cangelosi, and E. Coutinho, "The multiple voices of musical emotions: Source separation for improving music emotion recognition models and their interpretability," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020, pp. 310–317.

[6] S. Chowdhury, A. Vall, V. Haunschmid, and G. Widmer, "Towards explainable music emotion recognition: The route via mid-level features," *arXiv preprint arXiv:1907.03572*, 2019.

[7] M. Zentner, D. Grandjean, and K. R. Scherer, "Emotions evoked by the sound of music: characterization, classification, and measurement." *Emotion*, vol. 8, no. 4, p. 494, 2008.

[8] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[9] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer International Publishing, 2021.

[10] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *19th International Society for Music Information Retrieval Conference (ISMIR2018)*, 2018.

[11] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," *CoRR*, vol. abs/1703.09179, 2017. [Online]. Available: http://arxiv.org/abs/1703.09179

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] D. Yarotsky, "Universal approximations of invariant maps by neural networks," *CoRR*, 2018.

[14] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/