

Master thesis on Sound and Music Computing  
Universitat Pompeu Fabra

# Evaluation of state of the art for genre classification in large datasets

Vibhor Bajpai

**Supervisor:** Dmitry Bogdanov

**Co-Supervisor:** Alastair Porter

September 2018





Master thesis on Sound and Music Computing  
Universitat Pompeu Fabra

# Evaluation of state of the art for genre classification in large datasets

Vibhor Bajpai

**Supervisor:** Dmitry Bogdanov

**Co-Supervisor:** Alastair Porter

September 2018





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Conceptual background . . . . .	1
1.2	Motivation . . . . .	2
1.3	Objectives and structure of the report . . . . .	3
<b>2</b>	<b>State of the Art</b>	<b>4</b>
2.1	Classical Machine learning approach . . . . .	4
2.1.1	Low level feature extraction . . . . .	4
2.1.2	Machine learning classifier: Support Vector Machine . . . . .	5
2.1.3	Performance measures . . . . .	7
2.2	Datasets for genre classification . . . . .	9
<b>3</b>	<b>Datasets</b>	<b>12</b>
3.1	Jamendo song collection . . . . .	12
3.1.1	About Jamendo . . . . .	12
3.1.2	Song collection . . . . .	12
3.2	Jamendo dataset . . . . .	14
3.3	Rosamerica dataset . . . . .	15
3.4	LastFM dataset . . . . .	15
<b>4</b>	<b>Methodology</b>	<b>17</b>
4.1	Dataset creation . . . . .	17
4.1.1	Use of source tags provided in Jamendo . . . . .	17

4.1.2	Filtering songs based on audio length . . . . .	19
4.1.3	Choosing genre classes . . . . .	19
4.2	Genre classification steps and setup . . . . .	21
4.2.1	Low-level feature extraction . . . . .	22
4.2.2	Feature preprocessing and selection . . . . .	24
4.2.3	Classification using SVM . . . . .	25
4.3	Experiments . . . . .	26
4.3.1	Cross fold validation . . . . .	26
4.3.2	Cross collection evaluation . . . . .	26
<b>5</b>	<b>Results and discussion</b>	<b>29</b>
5.1	Tables and graphics . . . . .	29
5.1.1	5 fold cross validation . . . . .	29
5.1.2	Cross collection evaluation . . . . .	34
5.2	Discussion . . . . .	36
5.2.1	Significance of better feature selection . . . . .	36
5.2.2	Effect of album filter on cross-fold validation . . . . .	37
5.2.3	Dataset performance in cross-collection evaluation . . . . .	38
<b>6</b>	<b>Conclusions</b>	<b>39</b>
6.1	Conclusion . . . . .	39
6.2	Contributions . . . . .	39
6.3	Further Work . . . . .	39
	<b>List of Figures</b>	<b>41</b>
	<b>List of Tables</b>	<b>43</b>
	<b>Bibliography</b>	<b>44</b>

## Dedication

This thesis is dedicated to the three most important women in my life, my mother, grandmother, and my sister. You are the reason for all the good inside me.





## Acknowledgement

I would like to express my sincere gratitude to Dmitry Bogdanov, my thesis supervisor, who has been a great mentor and helped me throughout with his guidance and feedback. I would like to thank my co-supervisor Alastair Porter for his constant support, especially on the technology side. I would also like to thank Professor Xavier Serra, who has taught me a lot of what I know in Music technology and whose door was always open whenever I needed any help or guidance. I would like to thank Jordi Pons for his suggestions when I needed one. I am grateful to Sankalp Gulati and Shefali Bajpai who have provided valuable feedback and support during all times. And finally, I am grateful to Kushagra, Siddharth, and Manaswi for the endless discussions and brainstorming, which added to the joy of writing this thesis.



## Abstract

The goal of this thesis is to evaluate state of the art methods for genre classification on some popular genre datasets and provide an alternate dataset for the music community to use for the task of genre classification. Genre classification has been one of the main classification tasks in the MIR community due to its direct use in auto tagging of songs by the research community and the music industry alike. Companies like Spotify, YouTube, SoundCloud find it essential to tag songs based on genres since it is an important way to sort songs and gauge the listening styles of the users. It is, therefore, essential to study the features and models which might help in better classifying the songs based on various genres.

However, there aren't many quality datasets which have songs which are publicly available, balanced in terms of the number of genre classes present in them, and clean audio with longer durations. This thesis is an attempt to create models with better feature selection, and creating a dataset leveraging the publicly available Jamendo audio set for the task of genre classification. The presented models and dataset creation methods are provided with the aim to create better generalizable audio dataset.

Keywords: Genre classification;Jamendo



# Chapter 1

## Introduction

### 1.1 Conceptual background

Genre comes from the french word ‘genre’ which means ‘kind’ or ‘sort’.[1] Hence Genre classification in music means to identify similar characteristics of various musical pieces and classify them in a group called genre. These characteristics typically are related to the instrumentation, rhythmic structure, and harmonic content of the music. [2] Hence, music genres are categories that have arisen through a complex interplay of cultures, artists, and market forces to characterize similarities between musicians or compositions and organize music collections. [3]

These collections are not only organized by various music labels and music distribution companies, but also by end users who buy/listen to the music. It not only helps in organizing music in a better way, but also helps in providing better music recommendation by classifying similar music as per the play list of a user. However, many a times musical genres are subjective and thus the boundaries between genres still remain fuzzy as does their definition, making the problem of automatic classification a nontrivial task. [3]

## 1.2 Motivation

Genre classification forms a major component in the Music Information Retrieval tasks but a lack of good quality audio datasets has been a problem for the community. [4] A lot of datasets are not balanced i.e. they do not have equal number of songs in each genre classes. It is therefore recommended to have a dataset which is balanced for a robust training and testing. [5] Many of the datasets aren't publicly available, most of them being accessible only in MIR conferences and competitions like MIREX [6] therefore restricting the research in the field. The audio clips are generally short (less than 30 seconds) and there's a poor performance over cross collection analysis. [7]

Therefore it seemed to be an interesting and important topic to solve these shortcomings and thus, in this thesis we have created a balanced dataset, with longer audio length which is publicly available for research in genre classification. The fact that there was an openly available song collection available in form of Jamendo [8] with well tagged meta-data helped in creating such a dataset. The choice of choosing the genres equivalent to the contemporary datasets like GTZAN [2] and Free Music Archive [4] meant a lot of curating and cleaning of the song collection which was in itself a challenging task with plenty of tags to skim through and a lot of conflicts to take care of.

One of the main challenges was to resolve the conflict in tags of genres since a single audio song could be tagged using multiple genre labels and also to be sure if the labels were tagged correctly, it was essential to make use of human annotation methods like Amazon mechanical turk [9] and the tags directly labeled by the artists of the respective songs. Apart from that, the performance analysis using the benchmarks in the current genre classification techniques was also an important part of the thesis work. Providing a dataset consisting of songs with longer audio duration, with the aim of more data for any potential deep learning analysis was also a big motivation. [10]

## 1.3 Objectives and structure of the report

Main objectives and structure of the present study are:

- Review and analysis of current techniques for Genre classification task
- Creation of a balanced dataset for genre classification using publicly available songs from Jamendo song collection
- Performance analysis of the newly created dataset using the reviewed techniques, and machine learning models and classifiers
- Cross collection analysis of the curated dataset with other contemporary datasets

# Chapter 2

## State of the Art

In this section, two fields of study are reviewed. First, we present the classical machine learning approach for genre classification, then we discuss the contemporary datasets which are available for the same.

### 2.1 Classical Machine learning approach

The current machine learning approaches employ techniques to evaluate low level descriptors (spectral, time domain, tonal, rhythm) and statistics of these features (mean, median, variance, covariance) [11]. The low level descriptors are then used as features for machine learning classifiers such as Support Vector Machine [12] which then classify which genre a song/audio belongs to.

#### 2.1.1 Low level feature extraction

Low level feature extraction is one the fundamental steps in any Music Information Retrieval (MIR) task. The audio is first loaded using audio loaders like FFmpeg[13]/Libav[14] and then processed further. There are a lot of tools and libraries which can be used to extract low level features from an audio signal like Essentia [11], Librosa [15] etc. In this report, we make use of Essentia to extract the low level descriptors.



Essentia is an open-source C++ library for audio analysis and audio-based music information retrieval released under the Affero GPL license. It is based on various algorithms for extracting low level features which are contributed by more than 20 researches over a span of many years. It is a cross-platform library which can be executed on platforms like Mac OS x, Linux, and Windows. The library is also wrapped up in Python and can be used as a predefined executable extractor for various music descriptors, some of which are as follows:

- Spectral descriptors: the energy of the given frequency band or the set of bands of a spectrum, the Bark band energies of a spectrum, the Mel band energies, the Melfrequency cepstral coefficients, and the ERB band energies of a spectrum
- Tonal Descriptors: the pitch salience function of a signal, the estimation of the fundamental frequency of the predominant melody, the Harmonic Pitch-Class Profile (HPCP) of a spectrum (also called chroma features)
- Rhythm descriptors: the beat tracker based on the complex spectral difference feature, onset detection functions for the audio signal including HFC, spectral flux and Mel-bands based spectral flux
- Statistics: the mean, geometric mean, power mean, median of an array, and all its moments up to the 5th-order, its energy and the root mean square (RMS), flatness, crest and decrease of an array, typically used to characterize the spectrum, variance, skewness, kurtosis of a probability distribution, and a single Gaussian estimate for the given list of arrays (returns the mean array, its covariance and inverse covariance matrices)

### 2.1.2 Machine learning classifier: Support Vector Machine

After the low level descriptors are extracted, we make use of them to classify the audio they were extracted from in a meaningful way. There are a lot of machine learning classifiers available for the same like linear regression, random forest, support vector machine (SVM) etc. In this study, we make use of SVM for the classification.

Support vector machine (SVM) learning is a useful statistic machine learning technique that has been successfully applied in the pattern recognition area.

Suppose we are given a set of training data  $(x_1, x_2, x_3, \dots, x_n)$  and their class labels  $(y_1, y_2, y_3, \dots, y_n)$  where  $x_i \in R^n$  and  $y_i \in -1, +1$  and we want to separate the training data into two classes. If the data are linearly non-separable but non-linearly separable, the non-linear SVM classifier will be applied. The basic idea is to transform input vectors into a high dimensional feature space using non-linear transformation  $\Phi$  and then to do a linear separation in feature space. To construct a non-linear SVM classifier, inner product  $\langle x, y \rangle$  is replaced by a kernel function  $K(x, y)$ .

$$f(x) = \text{sgn}\left(\sum_{i=1}^l a_i y_i K(x_i, x) + b\right) \quad (2.1)$$

The SV algorithm can construct a variety of learning machines by use of different kernel functions. Three kinds of kernel functions are constantly used. They are:

- Polynomial kernel of degree  $d$

$$K(x, y) = (\langle x, y \rangle + 1)^d \quad (2.2)$$

- Radial basis function with Gaussian kernel of width  $c > 0$

$$K(x, y) = \exp(-\|x - y\|^2/c) \quad (2.3)$$

- Neural networks with tanh activation function

$$K(x, y) = \tanh(k \langle x, y \rangle + \mu) \quad (2.4)$$

where the parameters  $k$  and  $\mu$  are the gain and shift.

The low level features which are extracted using feature extraction libraries are used as input into the SVM classifier which then classifies the audio as one of the classes i.e. genres in this case. Machine learning classification is generally done in two stages, *training* and *testing*. In the training stage, the classifier learns which input parameters best represent a particular class by using previously annotated data i.e. ground truth. Generally speaking, more the number of examples for training, better the learning, hence more accurate is the classification. The testing stage helps to quantify the accuracy of the classification model by noting how many examples were correctly classified by the classifier.

The above mentioned method however, corresponds to supervised learning which is a classification method where ground truth data is already present to accurately quantify the performance of a classifier. There is another type of learning called unsupervised learning where the labels are not present and it is not easy to quantify the accuracy of the model. The genre classification task as such is a supervised learning task, where the genre labels are already present for the ground truth.

### 2.1.3 Performance measures

This section summarizes various performance measures of a classification model and how to interpret them. Once a model is built, it is important to quantify its performance using certain metrics so that a fair comparison could be made with other models. However, before listing the performance measures, let's understand some of the important terms related to them:

- True Positive (TP): These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. For example, if the actual class value states the genre is 'rock' and the predicted class value is also the same.
- True Negatives (TN): These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. For example, if the actual class value states the genre is not rock and the

predicted class value states the same.

- False Positives (FP): When actual class is no and predicted class is yes. For example, if the actual class value states the genre is not rock but the predicted class value states it is rock.
- False Negatives (FN): When actual class is yes but predicted class is no. For example, if the actual class states the genre is rock but the predicted class value states it is not.

Now that we have an understanding of these parameters, we can now see how various performance measures are evaluated using the same. There are four major performance metrics which are calculated:

1. Accuracy: Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.5)$$

2. Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = \frac{TP}{TP + FP} \quad (2.6)$$

3. Recall (Sensitivity): Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

$$Recall = \frac{TP}{TP + FN} \quad (2.7)$$

4. F-measure: F-measure is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F-measure is usually more useful than accuracy, especially if you have an uneven class distribution.

Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

$$Fmeasure = \frac{2 \cdot (Recall \cdot Precision)}{(Recall + Precision)} \quad (2.8)$$

## 2.2 Datasets for genre classification

Genre classification is one of the most popular tasks in MIR. Therefore, there has been a lot of research done to study various methods to improve classification models. An important aspect of this is to create datasets to drive the research further. This section summarizes some of the better known datasets in the MIR community for the task of genre classification.

Dataset	#Clips	Year	Access
Ballroom	698	2004	yes
GTZAN	1000	2002	yes
ISMIR 2004	1458	2004	yes
Unique	3115	2010	yes
USPOP	8752	2003	no
MagnaTagATune	25863	2009	yes
FMA	106574	2017	yes
MSD	1,000,000	2011	no
AcousticBrainz	3,514,717	2018	no

Figure 1: Datasets for Genre classification

Let's discuss some of the datasets mentioned in the table above:

1. Ballroom dataset: The Ballroom dataset was created for the audio description contest of ISMIR 2004. [16] [17] It was extracted from the website [www.ballroomdancers.com](http://www.ballroomdancers.com) at that time. The ballroom test-set contains 698 music excerpts of 30 seconds each, divided into 8 genres representing various Ballroom dances. The audio quality of the dataset is quite poor and the number of songs are also quite low. [18]
2. GTZAN: GTZAN [2] dataset is one of the most used datasets in music genre classification. [19] This dataset is composed of 1000 half-minute music audio excerpts singly labeled in ten genre categories. One of the main reasons for the popularity of this dataset is that it was one of the first datasets to be available for research publicly, however now it is widely believed that despite being a highly favored dataset for various researches, the use of the dataset should be discarded. [19]
3. ISMIR 2004: This dataset was created by Music Technology Group (MTG) [20] at UPF, Barcelona for ISMIR 2004 [16] genre classification task. The audio for the task was collected from Magnatune, [21] which contains a large amount of music licensed under Creative Commons licenses. It consists of 2 sets for training and development with 729 songs in each set and 6 genre classes with number of songs proportional to the songs present in those classes in Magnatune at that time.
4. USPOP2002 dataset: This dataset was created by Labrosa [22] and it consists of a total of 706 albums and 8764 tracks by 400 artists.
5. Magnatagatune: The MagnaTagATune dataset [23] consists of almost 26,000 audio tracks of 29 seconds duration each. Tracks are sampled with sampling rate 16 kHz, so bandwidth is limited to 8 kHz. Every sample is annotated in 189 categories with binary value.

6. Free Music archive (FMA) dataset: This dataset is created by using the dump of the Free Music Archive (FMA), an interactive library of legal audio downloads.[4] There are 3 types of datasets provided according to the number of audio files in it. They are:

- FMA small: 8,000 tracks of 30s, 8 balanced genres (GTZAN-like)
- FMA medium: 25,000 tracks of 30s, 16 unbalanced genres
- FMA large: 106,574 tracks of 30s, 161 unbalanced genres

FMA dump consists of 106,574 tracks from 16,341 artists and 14,854 albums, arranged in a hierarchical taxonomy of 161 genres.

7. Million Song Dataset (MSD): Million Song Dataset [24] is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. It contains:

- 1,000,000 songs/files
- 44,745 unique artists
- 43,943 artists with at least one term
- 515,576 dated tracks starting from 1922

Despite it's large size, the main drawback of MSD is the unavailability of the original audio tracks, hence the scope for novel acoustic representations is limited to those that have already been derived for the dataset.

8. AcousticBrainz: AcousticBrainz [25] is a joint initiative of Music Technology Group at UPF in Barcelona [20] and the MusicBrainz project. [26]. The dataset currently has 3,617,045 unique recordings. The project aims to crowd source acoustic information for all music in the world and to make it available to the public. This acoustic information describes the acoustic characteristics of music and includes low-level spectral information and information for genres, moods, keys, scales and much more.

# Chapter 3

## Datasets

This chapter describes the datasets that have been used for the research work and evaluation. It starts with the Jamendo dataset which was created using Jamendo song collection [8] which is a song repository for independent artists/musicians where they can upload and share their music for free. The following sections discuss about the Jamendo song collection, the Jamendo dataset which is the focus of this work and how the dataset was created from the song collection.

### 3.1 Jamendo song collection

#### 3.1.1 About Jamendo

Jamendo is a music website and a community of independent artists where musicians from around the world could upload their music for free. It was launched in 2005 in Luxembourg. The company describes itself as one of the world's largest digital service for free music.

#### 3.1.2 Song collection

Some statistics for the Jamendo song collection:

- Jamendo has around 201,634 songs which were available for download for this



work.

- Out of all the available songs, only 99,978 songs have a good metadata.
- A total number of 56324 songs that have been tagged by artists
- There are around 254 tags that have been used to tag the songs in the collection.
- Some of the genre tags in the collection are pop, rock, dubstep, hiphop, dance, electronic, jazz, country, ambient, house, rap, folk, blues, reggae, lounge, world, poprock, experimental, metal, indie, rnb, world, latin. [8] Out of these tags, electronic, pop, and rock have been used the most number of times.

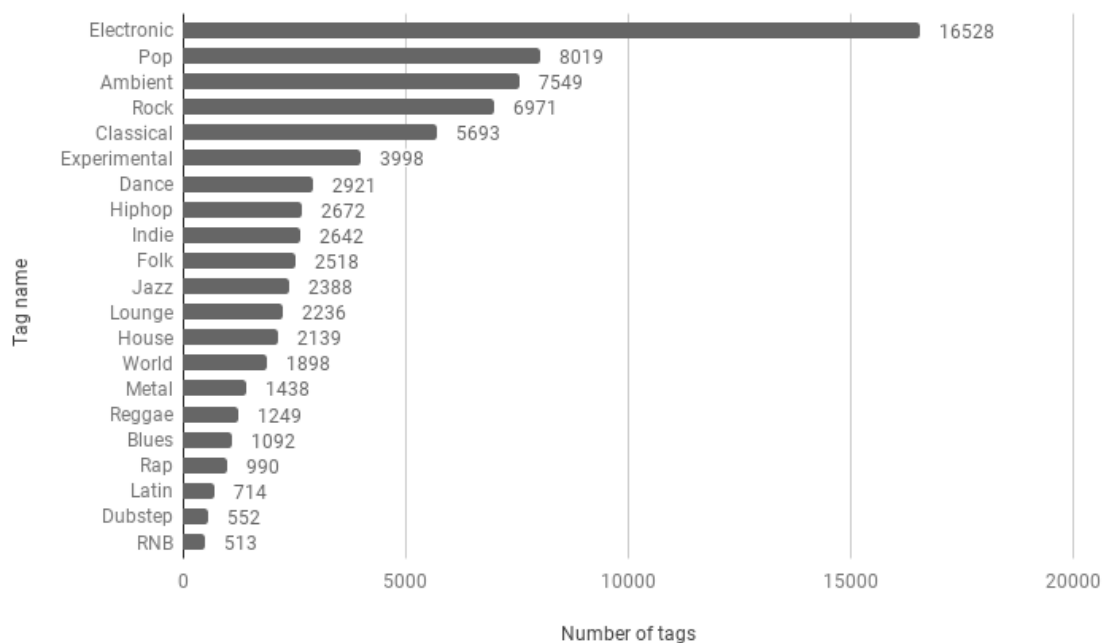


Figure 2: Common tags in Jamendo song collection

## 3.2 Jamendo dataset

A dataset was created for this thesis from the aforementioned Jamendo song collection. Out of all the available audio content in the Jamendo music collection, a total number of 5544 songs were selected representing 8 genres. Some statistics for the Jamendo dataset are as follows:

- There are total 5544 songs in the dataset.
- The dataset consists of 8 genre classes. The genres are as follows:
  1. Rock
  2. Classical
  3. Folk
  4. Jazz
  5. Metal
  6. Hip hop
  7. Reggae
  8. Dance
- The dataset is balanced i.e. there are equal number of songs in each genre class. Every genre class consists of 693 songs each.
- A total number of 3189 albums were chosen to create the song pool.
- The total number of 1503 artists were chosen.
- Each song is minimum 60 seconds long. The mean duration of the songs is 247 seconds and longest song has a duration of 980 seconds.

Further details about steps to create the dataset are mentioned in the methodology section.

### 3.3 Rosamerica dataset

Rosamerica is an in-house dataset created by Enric Guaus [27] at MTG [20] in UPF, Barcelona. This dataset was chosen for the cross collection [7] analysis alongside the Jamendo dataset. Some of the statistics for the Rosamerica dataset are:

- There are a total of 435 audio files in the dataset.
- The dataset consists of 8 classes, which are as follows:
  1. Rock
  2. Classical
  3. Jazz
  4. Dance
  5. Rhythm
  6. Hip hop
  7. Pop
  8. Speech
- The dataset is balanced with 54-55 audio files in each class
- There are total 337 artists for the songs in the dataset.
- The minimum duration of a song is 61 seconds and maximum duration is 888 seconds.

### 3.4 LastFM dataset

LastFM [28] is a song tagging dataset derived from Million song Dataset. [24] Although the original LastFM dataset consists of around 943,347 tracks matched with MSD, only selected song tracks and genres have been used for the purpose of this thesis. Some statistics regarding the customized LastFM dataset are:

- Number of songs used is 236,607 out of 943,347 tracks from the complete LastFM song collection.
- The genres used are:
  1. Rock
  2. Classical
  3. Hip hop
  4. Jazz
  5. Dance
- Dataset isn't balanced, i.e. there are different number of tracks in different genre classes.

# Chapter 4

## Methodology

This chapter discusses the proposed method and its implementation for the task of genre classification, creation of the Jamendo dataset, and cross collection [7] analysis using other datasets.

### 4.1 Dataset creation

The Jamendo dataset that was created for this thesis, is a subset of a much bigger Jamendo song collection. [8] The song collection has been used previously for evaluation for genre classification. [29] Since the song collection is huge (around 200,000 songs), it was necessary to create certain filtering schemes to arrive at a smaller, final dataset. Some of the techniques that were used to do the filtering of songs is as follows:

#### 4.1.1 Use of source tags provided in Jamendo

Jamendo makes use of a lot of third party genre tag verification for all the genre tags associated with the songs in its huge collection. Some of the sources are as follows:

- Artist: The artists who create the songs tag the genre of the audio they upload on Jamendo themselves. This is supposed to be a reliable indicator of

the accuracy of the tags because the tags come from the creators themselves and there is lesser scope of any error or misjudgment while tagging the audio.

- Amazon Mturk: Amazon Mechanical Turk (MTurk) is a crowdsourcing Internet marketplace enabling individuals and businesses (known as Requesters) to coordinate the use of human intelligence to perform tasks that computers are currently unable to do. [9] The idea is to assist computers with a human help in tasks which they are not able to perform due to various limitations. Since the tags are cross-verified by various individuals, it is also a good indicator of the correctness of tags that have been used for a song.
- BMAT: Barcelona Music Audio Technologies (BMAT) [30] is a music identification, monitoring, and fingerprinting company. It has created automatic annotations and tags for the Jamendo song collection. These are tags are less reliable as compared to others.
- Auto2: These are the tags created by Niland [31] which is a technology partner for Jamendo.

For the creation of the Jamendo dataset for this thesis and verification of the genre tags, only two sources, ‘artist’ and ‘mturk’, were considered since both are a reliable indicator of the tags. The filtering based on tag source was done as following:

1. Firstly, only those songs were considered which had both artist and mturk tag sources.
2. The genre tags annotated by both the sources were then compared, and then only those songs were considered which had a common genre tag.
3. In case there was no ‘mturk’ source tag present, only ‘artist’ source was considered for the tags.

This tag source filtering exercise left a pool of songs with sufficiently reliable genre tags for the dataset.

### 4.1.2 Filtering songs based on audio length

Apart from filtering the songs based on the source of the genre tags, the duration of the audio was also seemed to be a good criteria for filtering out the songs. Following is the chart showing the number of songs in

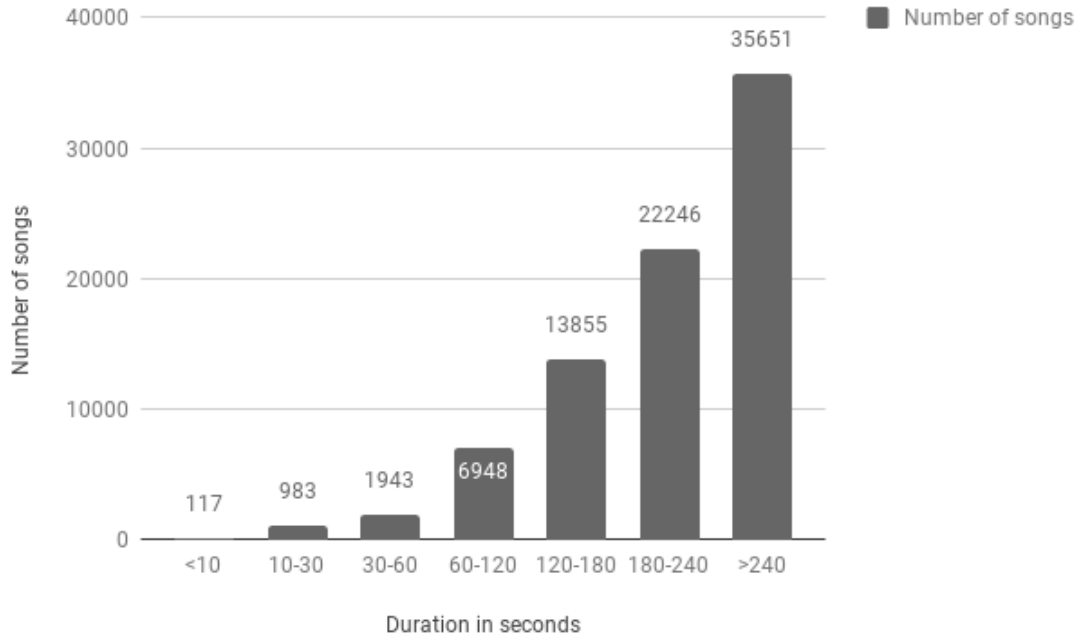


Figure 3: Duration of songs in Jamendo song collection

After listening to the songs of shorter duration ( $\sim 10$  seconds), it was found that a lot of clips were empty and many of them were not of good quality as well. Therefore, to avoid any low quality/empty content, only those songs were chosen which were at least 60 seconds long. That ensured that there were no short empty clips and also that the audio selected corresponded to full audio length songs.

### 4.1.3 Choosing genre classes

The genre classes were chosen based on the commonly used genres in contemporary datasets like GTZAN [2] and Rosamerica [27]. However, care was taken to choose genres which do not sound too similar. For example, dance and electronic are very similar sounding genres, and genres like pop have a very generic sound which

could sound similar to other genres like rock, dance, hip hop etc. Hence, for better understanding ‘Co-occurrence matrices’ was plotted for more commonly used genres from other datasets. A co-occurrence matrix represents the percentage of occurrence of one tag if the other tag is present as well.

This helped in understanding how the users/artists perceive various genres and how they use tags, thus helping to avoid too similar sounding genres.



Figure 4: Co-occurrence matrix showing prevalence of pop and electronic tags with other genre tags

We can see from the co-occurrence matrix in Figure 4 that the ‘pop’ tag occurs frequently with other tags like ‘rock’, ‘dance’, ‘folk’ etc. Similarly ‘electronic’ tag occurs frequently with ‘dance’, ‘reggae’, ‘pop’. Therefore these genre classes (i.e. pop and electronic) are not chosen for the final dataset.

In Figure 5 we can see that most of the genre tags have lesser co-occurrence with each other, therefore the genre boundaries seem to be better defined for this group of genre classes. The tag ‘metal’, however does has a high co-occurrence with the



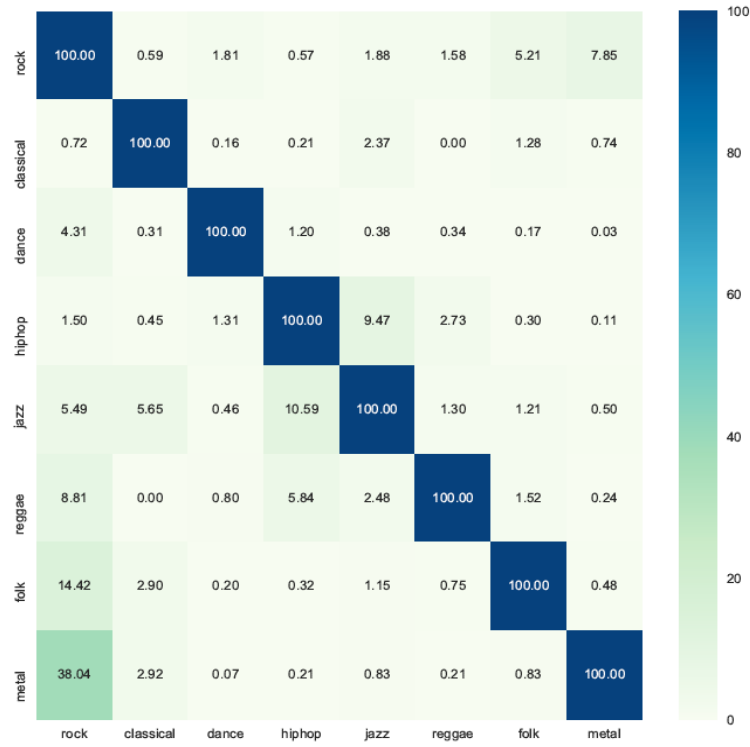


Figure 5: Co-occurrence matrix for the final Jamendo dataset

‘rock’ tag, but this seems intuitive because metal is supposed to have stemmed out of rock and the boundary between them is less well defined.

The Figure 6 summarizes how the Jamendo dataset was selected from the song collection. As can be seen in the figure, almost half the songs are discarded due to bad metadata. Then various filtering techniques as mentioned above were used to curate a dataset with 5544 songs. The final dataset is balanced and consists 8 classes, namely, Rock, Metal, Classical, Jazz, Hip-Hop, Dance, Folk, and Reggae were used. Each genre class consists of 693 songs each.

## 4.2 Genre classification steps and setup

The classification and evaluation setup was inspired from the one being used in AcousticBrainz. [25] It consists of three essential steps:

1. Feature extraction from the audio
2. Feature selection of the extracted features

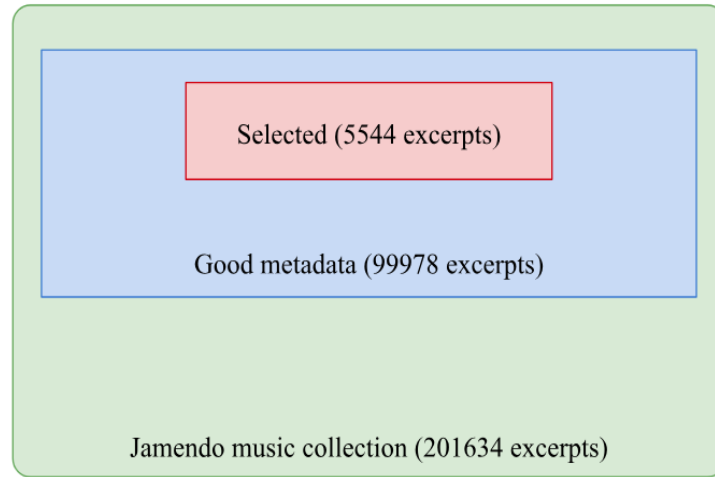


Figure 6: Diagrammatic representation of selecting songs for the Jamendo dataset from the song collection

### 3. Classification using machine learning classifier

Let us discuss the steps in detail:

#### 4.2.1 Low-level feature extraction

The audio from the dataset is loaded and then low level features are extracted using Essentia. [11] Apart from the timbral, rhythmic, and other low level features, the statistics of those features are also computed. The algorithms compute statistics over an array of values, or some kind of aggregation. These algorithms allow to compute:

- the mean, geometric mean, power mean, median of an array, and all its moments up to the 5th-order, its energy and the root mean square (RMS).
- variance, skewness, kurtosis of a probability distribution, and a single Gaussian estimate for the given list of arrays (returns the mean array, its covariance and inverse covariance matrices)

It is well established that feature statistics like mean, median, variance, and covariance [32] can be used for genre classification. Therefore in this study, we consider

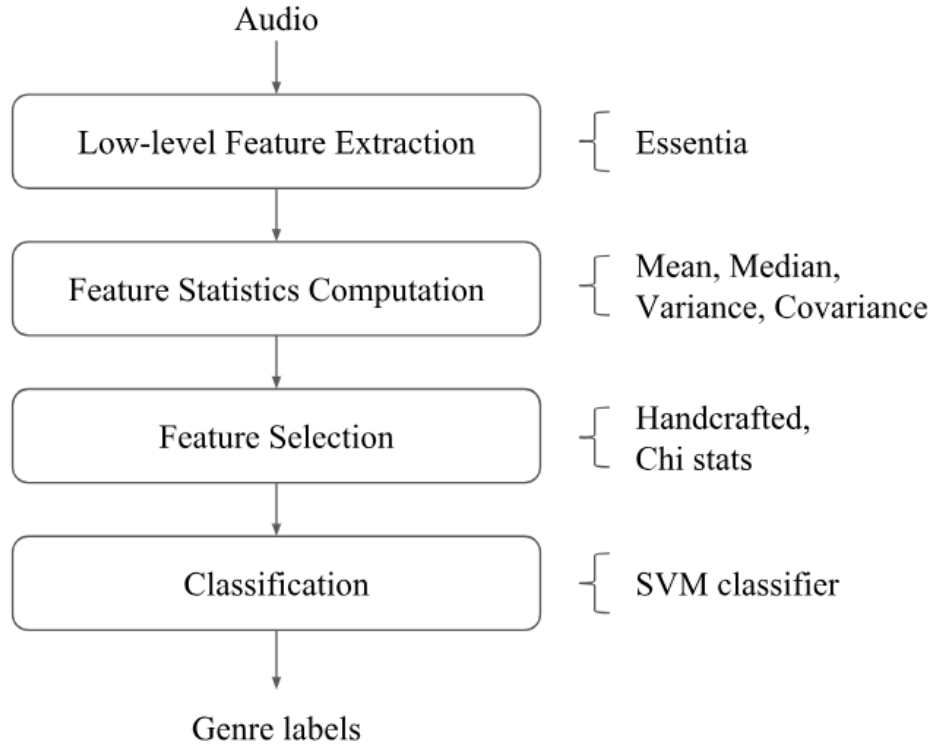


Figure 7: Flowchart for the steps involved in genre classification

different combinations of these 4 statistics of features to understand how useful these combinations are to do classify different genre classes. We also consider certain

The mean and variance are calculated by using the following equations:

$$Mean(\mu) = \frac{1}{N} \sum_{n=1}^N X_n \quad (4.1)$$

$$Std(\sigma) = \sqrt{\frac{\sum_{n=1}^N (X_n - \mu)^2}{N}} \quad (4.2)$$

Covariance is measured between two features. The aim of considering the covariance is usually to see if there is any mutual relationship between the features. It is useful to measure the polarity and the degree of the correlation between two features. The

covariance of two features  $X_n$  and  $Y_n$  in a music piece is given as:

$$Cov(X_n, Y_n) = \frac{1}{N} \sum_{n=1}^N (X_n - \mu_X)(Y_n - \mu_Y) \quad (4.3)$$

### 4.2.2 Feature preprocessing and selection

Essentia extracts a total of 2652 features and some non-useful features like ‘beats per minute’ and ‘metadata’. Before we proceed, we drop the bpm and metadata features. We also need to do certain conversions for some features so that they can be used by the classifier properly. The tonal features like chords and keys which are mentioned as musical notes like ‘a’, ‘b’ etc. need to be converted into corresponding numerical representation using one-hot encoding.

Feature normalization is also done using Quantile transformation. After these preprocessing steps, feature selection is done and 5 sets of features are created for further experiments:

1. Feature\_set\_basic: Consists of mean, median, variance, and tonal features like scale and key. Total features selected are 701.
2. Feature\_set\_custom: Consists of mean, median, variance, covariance, and tonal features like scale and key. Total features selected are 1377.
3. Feature\_set\_rare: Consists of mean, covariance, and tonal features like scale and key. Total features selected are 953.
4. Feature\_set\_nobands: Excludes erbbands, barkbands, energybands, and melbands. Total features selected are 326.
5. Feature\_set\_lowlevel: Only features marked as ‘lowlevel’ in the extractor are selected. Tonal features are ignored. Total features selected are 487.

From these 5 feature sets, chi squared statistics [33] was used to select k-best features by removing the redundant features from the feature set to create 13 feature

sets.

Feature set name	Number of features selected
f1	custom_1377
f2	custom_1050
f3	custom_900
f4	custom_850
f5	basic_701
f6	basic_650
f7	basic_600
f8	rare_952
f9	rare_900
f10	lowlevel_487
f11	lowlevel_400
f12	nobands_326
f13	nobands_300

Table 1: Feature set names and corresponding number of features

In this table ‘custom\_1050’ means a total number of 1050 features were selected using ch-squared statistics from the ‘Feature\_set\_custom’ 2 set.

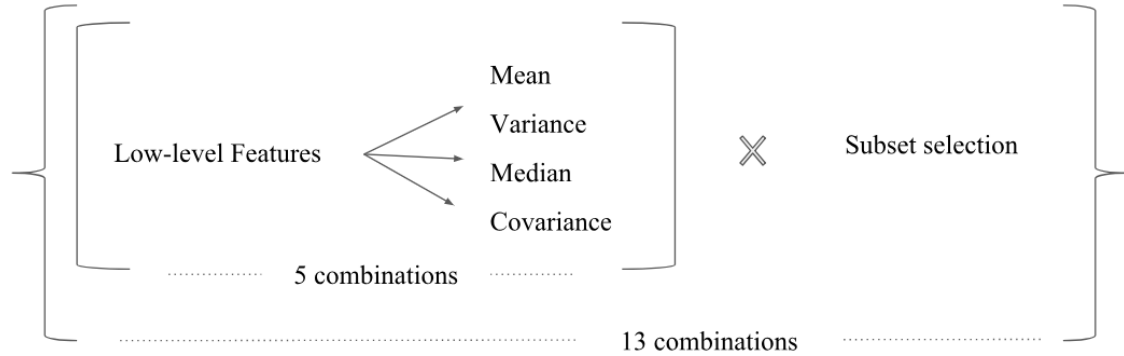


Figure 8: Creating 13 feature sets from the extracted features.

### 4.2.3 Classification using SVM

After the preprocessing and feature selection steps, classification using SVM as the classifier was done. The input to the classifier are the feature sets and the output

is the predicted class for the song for which the feature sets were extracted. The hyper-parameter tuning was done using grid search.

In the next section we mention the experiments that were done with this setup.

## 4.3 Experiments

Following experiments were carried out:

### 4.3.1 Cross fold validation

Cross fold validation experiments were carried out for both Jamendo [8] and Rosamerica datasets. The basic procedure included:

1. 5 fold cross validation repeated 10 times
2. Experiments done using album filter (for Jamendo dataset) and without album filtering (for both Jamendo and Rosamerica).
3. Feature normalisation was done using Quantile transformation.
4. The classifier used was SVM and hyper parameter tuning was done using gridsearch.
5. Each of the 13 feature sets were used for this part of experiment.

### 4.3.2 Cross collection evaluation

Cross collection [7] evaluation was done using Jamendo, Rosamerica, and LastFM datasets. Since Jamendo and Rosamerica have only 5 genre classes common, cross collection analysis was done with only those 5 classes. The setup was as follows:

- Jamendo as test set and Rosamerica as training set.
- Rosamerica as test set and Jamendo as training set.
- LastFM as test set and Rosamerica as training set.

- LastFM as test set and Jamendo as training set.

Note that we shall refer to Jamendo dataset as  $D_{Jam}$ , Rosamerica as  $D_{Rosa}$ , and LastFM dataset as  $D_{LFM}$ .

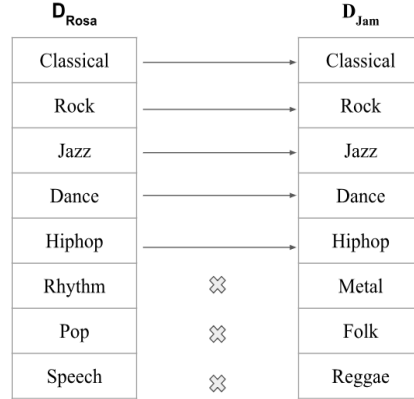


Figure 9: Genre class mapping between Rosamerica and Jamendo dataset

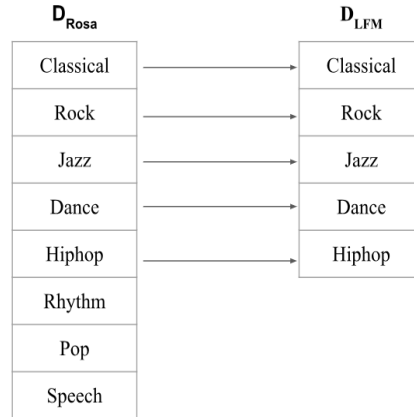


Figure 10: Genre class mapping between Rosamerica and LastFM dataset

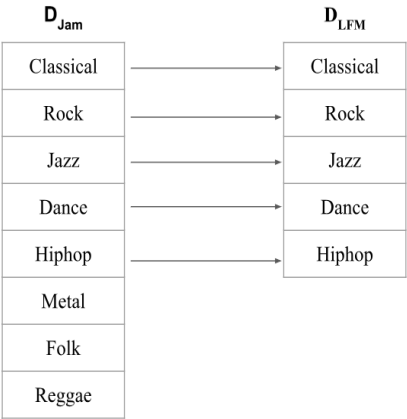


Figure 11: Genre class mapping between Jamendo and LastFM dataset



# Chapter 5

## Results and discussion

This chapter contains results for the experiments described in Section 4.3. The datasets used for 5 fold cross validation are Jamendo and Rosamerica. The datasets for cross-collection evaluation are Jamendo, Rosamerica, and LastFM.

### 5.1 Tables and graphics

#### 5.1.1 5 fold cross validation

The following tables present the mean accuracy for the  $D_{Jam}$  and  $D_{Rosa}$  datasets for all the 13 feature sets mentioned in table 1. (Mean accuracy here means total correct predictions/total number of samples as mentioned in Section 2.1.3)

Feature set name	Mean Accuracy
f1	<b>75.67</b>
f2	75.18
f3	74.51
f4	74.38
f5	73.90
f6	73.89
f7	74.16
f8	72.81
f9	73.78
f10	69.96
f11	70.60
f12	72.63
f13	72.03

Table 2: Mean accuracy values for each feature set for Jamendo dataset

Feature set name	Mean Accuracy
f1	<b>88.21</b>
f2	<b>89.12</b>
f3	<b>89.40</b>
f4	<b>89.53</b>
f5	87.91
f6	87.63
f7	88.03
f8	86.75
f9	86.23
f10	86.09
f11	86.07
f12	85.26
f13	85.47

Table 3: Mean accuracy values for each feature set for Rosamerica dataset

Feature	Accuracy without album filtering	Accuracy with album filtering
f1	75.67	72.77
f2	75.18	71.92
f3	74.51	71.09
f4	74.38	71.01
f7	74.16	71.79

Table 4: Mean accuracy values for the 5 best performing feature sets for Jamendo dataset with and without album filter

Following are the box plots and error analysis (confusion matrices) for 5 fold cross validation for Jamendo ( $D_{Jam}$ ) and Rosamerica ( $D_{Rosa}$ ) datasets.

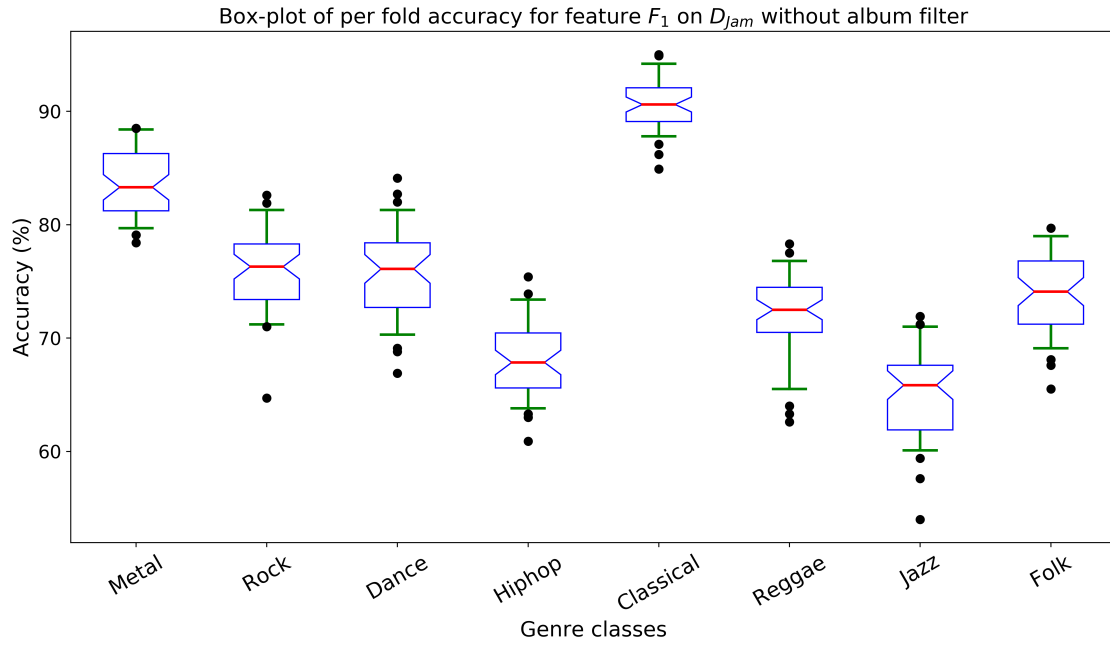


Figure 12: Box plot for  $D_{Jam}$  without album filter

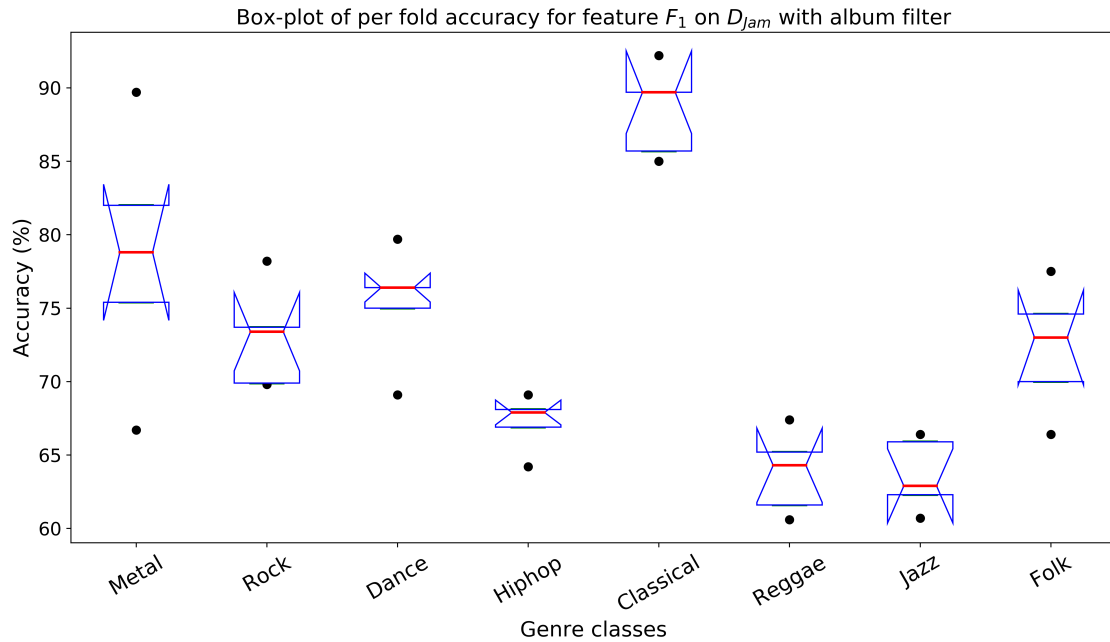
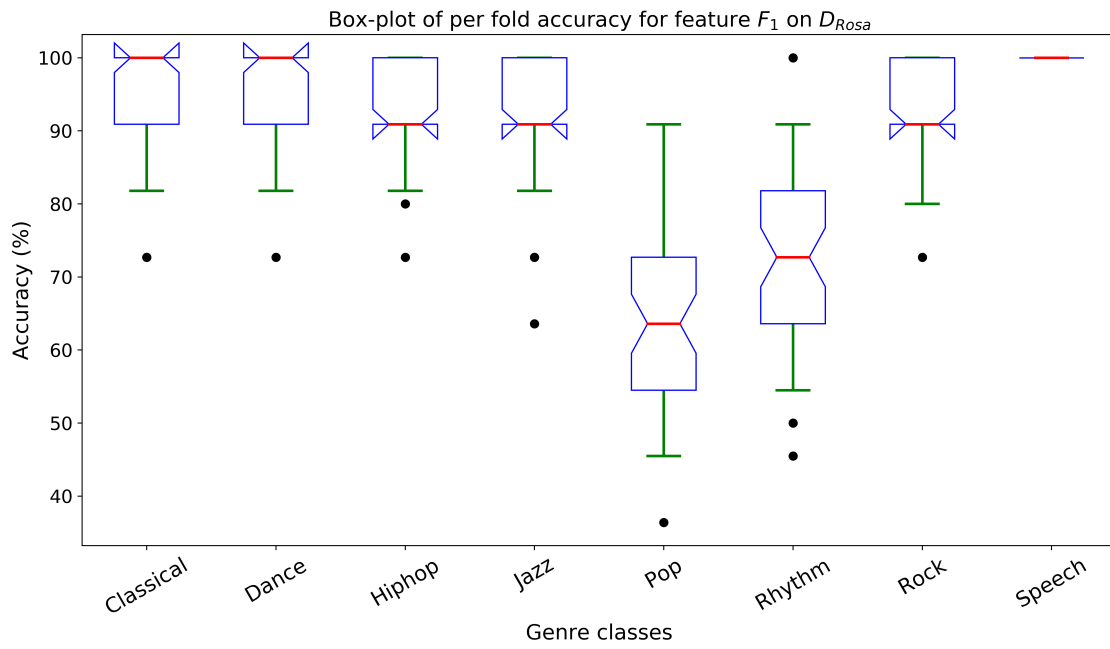
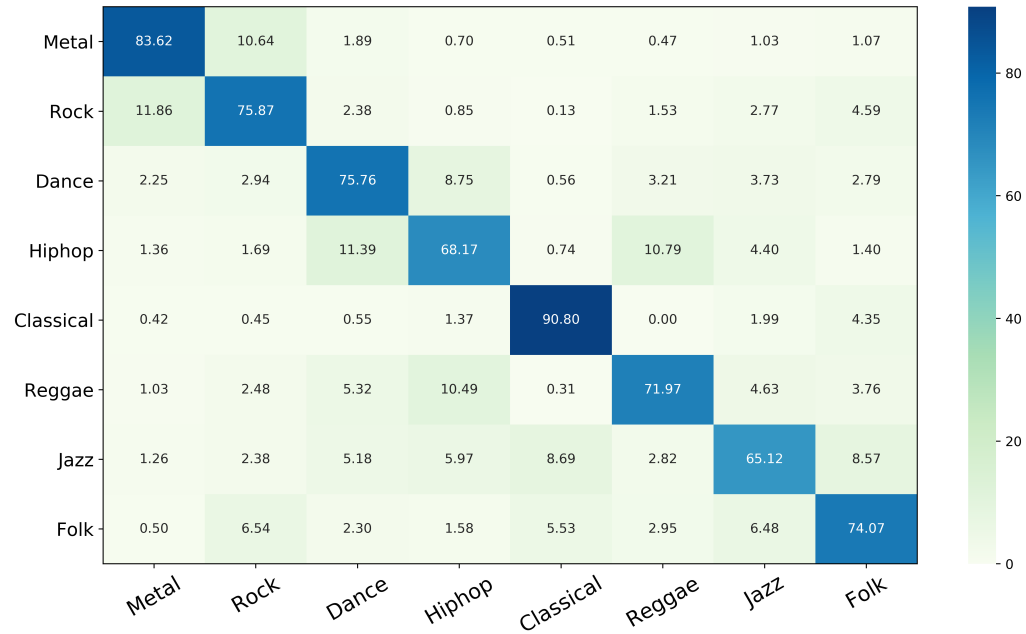
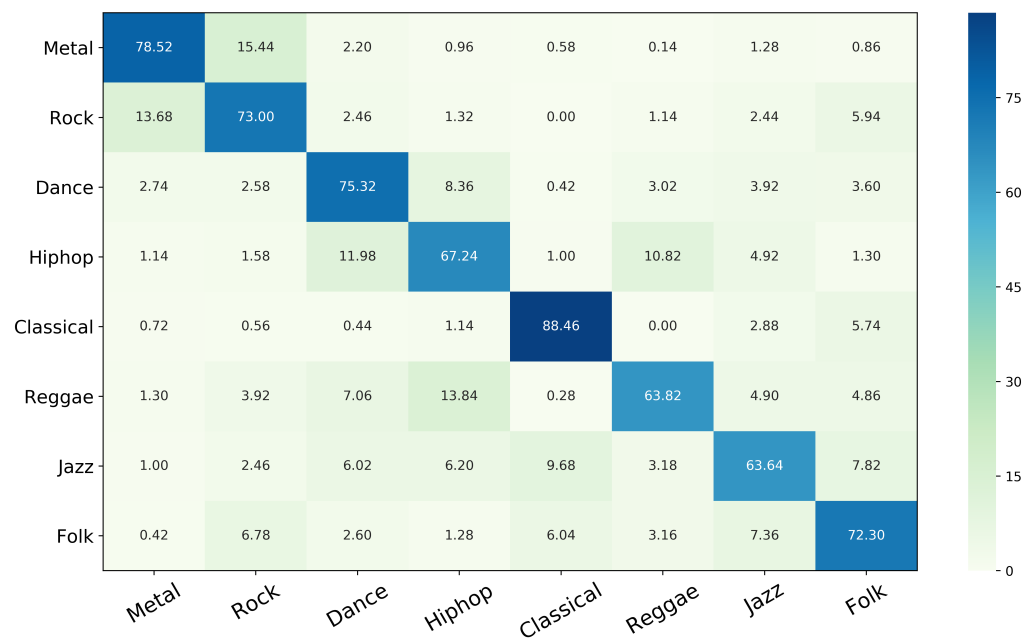


Figure 13: Box plot for  $D_{Jam}$  with album filter

Figure 14: Box plot for  $D_{Rosa}$ Figure 15: Confusion matrix for  $D_{Jam}$  without album filter

Figure 16: Confusion matrix for  $D_{Jam}$  with album filter

### 5.1.2 Cross collection evaluation

The following table present the cross collection evaluation results for Jamendo, Rosamerica, and LastFM datasets. Normalised accuracy means that in case of imbalanced testing set, the accuracies are normalised so as to give equal weightage to each class in the testing dataset.

		Test dataset		
Training dataset		$D_{Jam}$	$D_{Rosa}$	$D_{LFM}$
	$D_{Jam}$	-	<b>86.76</b>	43.03 <b>62.80*</b>
	$D_{Rosa}$	52.09	-	36.27 <b>54.80*</b>

\*Normalized mean accuracy

Figure 17: Cross collection evaluation result



Figure 18: Confusion matrix for  $D_{Jam}$  as training set and  $D_{Rosa}$  as testing set

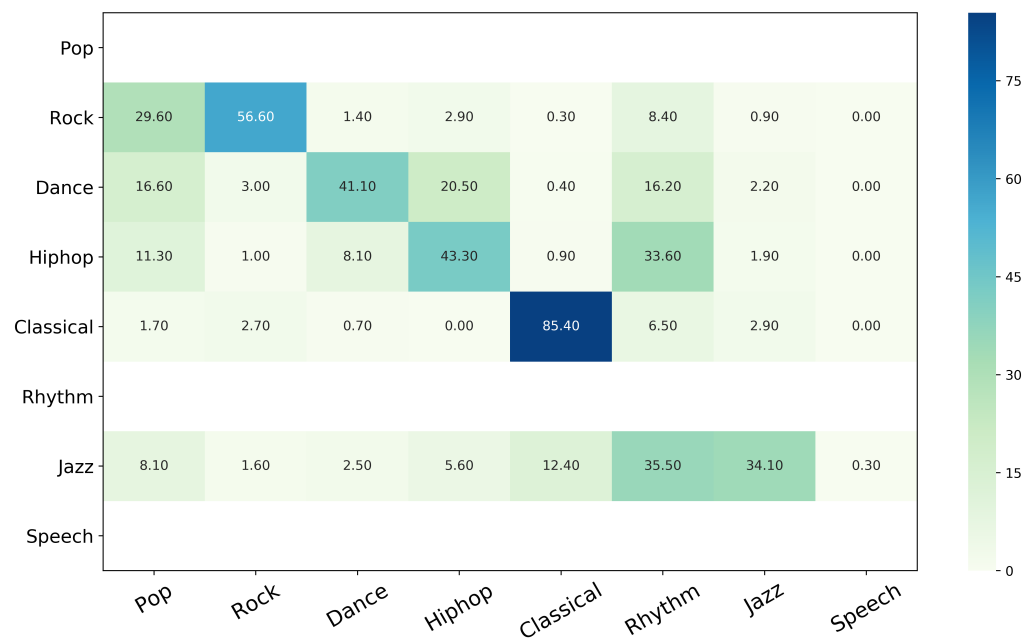
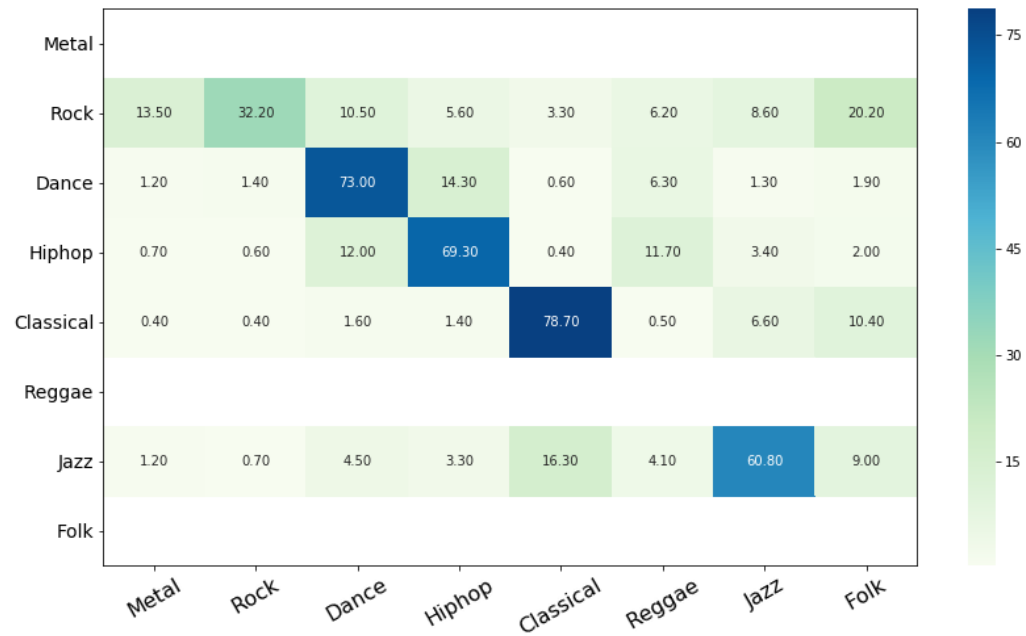
Figure 19: Confusion matrix for  $D_{Rosa}$  as training set and  $D_{Jam}$  as testing setFigure 20: Confusion matrix for  $D_{Jam}$  as training set and  $D_{LFM}$  as testing set



Figure 21: Confusion matrix for  $D_{Rosa}$  as training set and  $D_{LFM}$  as testing set

## 5.2 Discussion

### 5.2.1 Significance of better feature selection

We can see from the Table 2 and Table 3 that feature set consisting of mean, median, variance, and covariance, performs better than other feature sets in statistically significant way. This builds a case for better feature selection for statistics of features. Also we can note that the mean accuracies for Rosamerica dataset are better than Jamendo dataset. This can be explained by the fact that  $D_{Ros}$  is a significantly smaller dataset than  $D_{Jam}$  (435 songs in  $D_{Ros}$  as compared to 5544 songs of  $D_{Jam}$ ). Also the fact that the Rosamerica dataset was well curated and each song was well chosen after listening by a professional musicologist [27] helps in obtaining a better dataset. It was not possible to do the same for the Jamendo dataset due to its large size.



### 5.2.2 Effect of album filter on cross-fold validation

We can see in the Table 4 that the Jamendo dataset has a slightly better accuracy without album filter as compared to when the album filter was applied. Album filter means that the training and testing folds cannot have any songs from the same albums. This is to ensure that the genre classification does not become an album classification task. This can be better understood with the following figure:

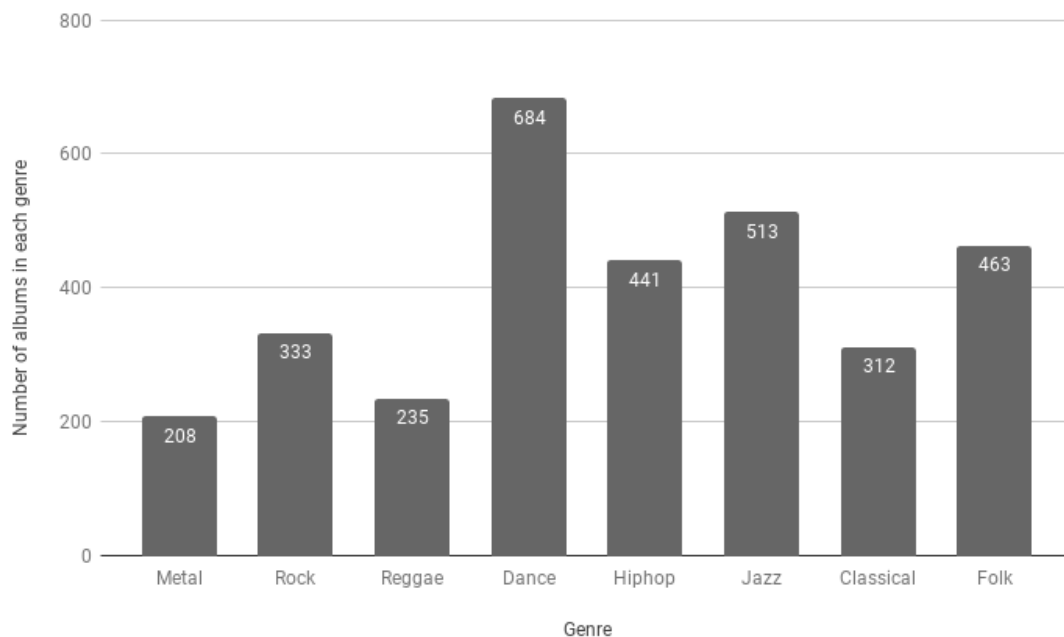


Figure 22: Number of albums in Jamendo dataset for each genre class

We can see from the Figure 22 the number of albums for each genre class in the Jamendo dataset. We see that metal and reggae have the least number of albums. Therefore when album filter is applied, there is a significant drop in accuracies for the same. The drop in accuracies for these particular genre classes can be seen in the Figure 15 and Figure 16.

### 5.2.3 Dataset performance in cross-collection evaluation

We can see from the Table 17 that Jamendo performs significantly better than Rosamerica dataset in Cross performance evaluation. It not only performs better with  $D_{LFM}$  as testing set and  $D_{Jam}$  as training set as compared to when  $D_{Rosa}$  is used as training set, it also performs better at classifying  $D_{Rosa}$  as compared to how  $D_{Rosa}$  classifies  $D_{Jam}$ .

Therefore, it seems obvious that Jamendo dataset has a far better performance in cross collection evaluation as compared to Rosamerica dataset. It is therefore more generalizable dataset for cross collection evaluation.

# Chapter 6

## Conclusions

### 6.1 Conclusion

The proposed dataset  $D_{Jam}$  performs well over cross collection evaluation with other datasets. It also holds up well during cross fold validation. It can therefore be used as a reliable dataset for studies in genre classification tasks, thus adding to the current state of the work in this field. The proposed model using statistics of features with mean, median, variance, covariance works well over different datasets and thus could be used for better classification accuracies.

### 6.2 Contributions

Current working model and code repository can be found at <https://github.com/vibhorbajpai/masterthesis>. The link to the dataset is also provided in the given url.

### 6.3 Further Work

This study can be seen as a preliminary research at providing the MIR community with an option to use another dataset for its genre classification tasks. The MIR community is moving towards deep learning slowly for better classification strategies

and this study explores the models using handcrafted low level features. It is therefore important to experiment with various deep learning models using this dataset and thus device better classification models for the same.

# List of Figures

1	Datasets for Genre classification . . . . .	9
2	Common tags in Jamendo song collection . . . . .	13
3	Duration of songs in Jamendo song collection . . . . .	19
4	Co-occurrence matrix showing prevalence of pop and electronic tags with other genre tags . . . . .	20
5	Co-occurrence matrix for the final Jamendo dataset . . . . .	21
6	Diagrammatic representation of selecting songs for the Jamendo dataset from the song collection . . . . .	22
7	Flowchart for the steps involved in genre classification . . . . .	23
8	Creating 13 feature sets from the extracted features. . . . .	25
9	Genre class mapping between Rosamerica and Jamendo dataset . . .	27
10	Genre class mapping between Rosamerica and LastFM dataset . . .	27
11	Genre class mapping between Jamendo and LastFM dataset . . . .	28
12	Box plot for $D_{Jam}$ without album filter . . . . .	31
13	Box plot for $D_{Jam}$ with album filter . . . . .	31
14	Box plot for $D_{Rosa}$ . . . . .	32
15	Confusion matrix for $D_{Jam}$ without album filter . . . . .	32
16	Confusion matrix for $D_{Jam}$ with album filter . . . . .	33
17	Cross collection evaluation result . . . . .	34
18	Confusion matrix for $D_{Jam}$ as training set and $D_{Rosa}$ as testing set . .	34
19	Confusion matrix for $D_{Rosa}$ as training set and $D_{Jam}$ as testing set . .	35
20	Confusion matrix for $D_{Jam}$ as training set and $D_{LFM}$ as testing set .	35
21	Confusion matrix for $D_{Rosa}$ as training set and $D_{LFM}$ as testing set .	36

22	Number of albums in Jamendo dataset for each genre class . . . . .	37
----	--	----

# List of Tables

1	Feature set names and corresponding number of features . . . . .	25
2	Mean accuracy values for each feature set for Jamendo dataset . . . .	30
3	Mean accuracy values for each feature set for Rosamerica dataset . .	30
4	Mean accuracy values for the 5 best performing feature sets for Ja- mendo dataset with and without album filter . . . . .	30

# Bibliography

- [1] Wikipedia. Genre. <https://en.wikipedia.org/wiki/Genre>.
- [2] Tzanetakis, G. & Cook, P. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* **10**, 293–302 (2002).
- [3] Scaringella, N., Zoia, G. & Mlynek, D. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine* **23**, 133–141 (2006).
- [4] Benzi, K., Defferrard, M., Vandergheynst, P. & Bresson, X. FMA: A dataset for music analysis. *CoRR* **abs/1612.01840** (2016). URL <http://arxiv.org/abs/1612.01840>. 1612.01840.
- [5] Sousa, J., Pereira, E. & Veloso, L. A robust music genre classification approach for global and regional music datasets evaluation (2016).
- [6] Mirex. [http://www.music-ir.org/mirex/wiki/MIREX\\_HOME](http://www.music-ir.org/mirex/wiki/MIREX_HOME).
- [7] Bogdanov, D., Porter, A., Herrera, P. & Serra, X. Cross-collection evaluation for music classification tasks. In *ISMIR* (2016).
- [8] Jamendo. <https://www.jamendo.com/>.
- [9] Amazon mechanical turk. <https://www.mturk.com/>.
- [10] Pons, J., Lidy, T. & Serra, X. Experimenting with musically motivated convolutional neural networks. *2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI)* 1–6 (2016).



- [11] Bogdanov, D. *et al.* Essentia: An audio analysis library for music information retrieval. In *ISMIR* (2013).
- [12] Xu, C., Maddage, N. C., Shao, X., Cao, F. & Tian, Q. Musical genre classification using support vector machines. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 5, V-429 (2003).
- [13] Ffmpeg. <http://www.ffmpeg.org/>.
- [14] Libav. <https://www.libav.org/>.
- [15] Librosa. <http://librosa.github.io/librosa/>.
- [16] Cano, P. *et al.* ISMIR 2004 Audio Description Contest. *Tech. Rep. MTG-TR-2006-02, Universitat Pompeu Fabra* (2006).
- [17] Ismir 2004. <http://ismir2004.ismir.net/>.
- [18] Marchand, U. & Peeters, G. The extended ballroom dataset. late-breaking demo session of the 17th international society for music information retrieval conference. In *ISMIR* (2016).
- [19] Sturm, B. L. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. *CoRR* **abs/1306.1461** (2013). URL <http://arxiv.org/abs/1306.1461>. 1306.1461.
- [20] Mtg. <https://www.upf.edu/web/mtg>.
- [21] Magnatune. <http://magnatune.com/>.
- [22] Labrosa. <https://labrosa.ee.columbia.edu/projects/musicsim/usp2002.html>.
- [23] Magnatagatune. <http://mirg.city.ac.uk/codeapps/the-magnatagatune-dataset>.
- [24] Bertin-Mahieux, T., P. W. Ellis, D., Whitman, B. & Lamere, P. The Million Song Dataset. 591–596 (2011).

- [25] Acousticbrainz. <https://acousticbrainz.org/>.
- [26] Musicbrainz. <https://musicbrainz.org/>.
- [27] Guaus, E. *Audio content processing for automatic music genre classification: descriptors, databases, and classifiers*. Ph.D. thesis, Universitat Pompeu Fabra (2009).
- [28] Lastfm. <https://labrosa.ee.columbia.edu/millionsong/lastfm>.
- [29] Kleć, M. Evaluation of Jamendo Database as Training Set for Automatic Genre Recognition. In Sombattheera, C., Loi, N. K., Wankar, R. & Quan, T. (eds.) *Multi-disciplinary Trends in Artificial Intelligence*, 296–305 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012).
- [30] Bmat. <https://www.bmat.com/home>.
- [31] Niland. <http://niland.io/technology>.
- [32] Kaji Baniya, B., Lee, J. & Li, Z.-N. Audio feature reduction and analysis for automatic music genre classification **2014**, 457–462 (2014).
- [33] Chi2 statistics. [http://scikit-learn.org/stable/modules/generated/sklearn.feature\\_selection.SelectKBest.html](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html).