Thomas Rönnberg

Classification of Heavy Metal Subgenres with Machine Learning

Master's Thesis in Information Systems
Supervisors: Dr. Markku Heikkilä
Asst. Prof. József Mezei
Faculty of Social Sciences, Business and Economics
Åbo Akademi University

Åbo 2020

Subject: Information Systems	
Writer: Thomas Rönnberg	
Title: Classification of Heavy Metal Subgenres with Machine Learning	
Supervisor: Dr. Markku Heikkilä	**Supervisor:** Asst. Prof. József Mezei

Abstract:
The music industry is undergoing an extensive transformation as a result of growth in streaming data and various AI technologies, which allow for more sophisticated marketing and sales methods. Since consumption patterns vary by different factors such as genre and engagement, each customer segment needs to be targeted uniquely for maximal efficiency. A challenge in this genre-based segmentation method lies in today's large music databases and their maintenance, which have shown to require exhausting and time-consuming work. This has led to automatic music genre classification (AMGC) becoming the most common area of research within the growing field of music information retrieval (MIR). A large portion of previous research has been shown to suffer from serious integrity issues. The purpose of this study is to re-evaluate the current state of applying machine learning for the task of AMGC. Low-level features are derived from audio signals to form a custom-made data set consisting of five subgenres of heavy metal music. Different parameter sets and learning algorithms are weighted against each other to derive insights into the success factors. The results suggest that admirable results can be achieved even with fuzzy subgenres, but that a larger number of high-quality features are needed to further extend the accuracy for reliable real-life applications.

Keywords: Artificial Intelligence, Machine Learning, Deep Learning, Supervised Learning, Predictive Analytics, Data Science, Data Mining, Music Information Retrieval, Automatic Music Genre Classification, Digital Signal Processing, Audio Signal Processing, Computational Musicology, Spectrograms, Heavy Metal

Date:	**Number of pages:** 120

Table of Contents

List of Acronyms

AI = Artificial Intelligence

AMGC = Automatic Music Genre Classification

ANN's = Artificial Neural Networks

CNN's = Convolutional Neural Networks

dB = Decibels

DCT = Discrete Cosine Transform

DFT = Discrete Fourier Transform

FN = False Negative

FP = False Positive

Hz = Hertz

HPSS = Harmonic-Percussive Source Separation

K-Fold CV = K-Fold Cross-Validation

KNN = K-Nearest Neighbors

LOOCV = Leave-One-Out-Cross-Validation

MFCC's = Mel-frequency Cepstral Coefficients

MIDI = Musical Instrumental Digital Interface

MIR = Music Information Retrieval

PCA = Principle Component Analysis

RMSE = Root-Mean-Squared Energy

RNN's = Recurrent Neural Networks

STFT = Short-Time Fourier Transform

SVM's = Support Vector Machines

TN = True Negative

TP = True Positive

ZCR = Zero-crossing Rate

1. Introduction

1.1 Background

The world around us is undergoing a fundamental change. A report by International Data Corporation (2017) discusses that data have become a critical aspect of human life during the past three decades, and they have changed our perception about entertainment, education, people, business and society. This ongoing growth in data, sometimes even called the era of Big Data, has not only come with new devices and forms of user experience, but also with new challenges. One of the primary challenges is concretized through a global survey by The Economist Intelligence Unit (2015). According to the survey, the proportion of companies that have come to regard data as corporate assets for strategic decision-making, is steadily increasing. As data grow in volume and variety, the number of companies unable to extract meaningful knowledge from their data is also increasing. Johnson (2018) explains that the difficulty in data-related value creation across various industries has in turn led to the spread of different forms of artificial intelligence (AI) technologies, such as predictive analytics and machine learning.

In his prologue, Marsland (2015) briefly introduces the multidimensionality of machine learning by stating that machine learning lies on the boundary of several academic fields, such as computer science, statistics, mathematics and engineering. Murphy (2012, 1) defines machine learning as a set of methods that can automatically detect patterns in data, and then use these uncovered patterns in order to predict future data, or to perform other kinds of decision-making under uncertainty. Moreover, a subfield of machine learning, which is often separately referred to as deep learning, has in recent years challenged the traditional machine learning methods. McClelland (2017) explains that deep learning is one of many approaches to machine learning. Deep learning is inspired by the structure and function of the brain, namely the interconnecting of many neurons, which results in algorithms that mimic the biological structure of the brain.

To date, these AI technologies have mainly seen use cases in industries such as healthcare, finance, manufacturing and retail, as exemplified by Johnson (2018). However, Marin (2018) explains that they have in recent years also gained a foothold within the music industry. Audio and music-facing technology companies such as Shazam and SoundHound have successfully utilized AI technologies that analyze large databases of songs by using spectrograms to measure various frequencies, which in turn enables the identification of a

song being played. In addition to this, Zhang (2018) clarifies that AI has also shown signs of success in creative practice through automatic music composition. Examples of this include IBM's Watson Beat, Google's Magenta, Sony's Flow Machines and the startup company AIVA.

While musical innovations are being developed with AI technologies, the structure of the music industry is also affected by AI technologies in two primary ways. Firstly, Delgado (2018) explains that AI technologies have allowed the creation of a bridge between sales and marketing by enabling the management of people, processes and partnerships of businesses on a micro scale. Secondly, they have enabled the possibility of generating information concerning questions that used to rely on solely subjective assumptions, such as why certain artists are more popular than others, which songs are more likely to become hits, or which genres are trending.

Marin (2018) concurs on the previously explained by establishing that AI-enabled data is turning the whole music industry into a more sophisticated one. Particularly for streaming services such as Spotify and YouTube, this AI-driven consumer data could be the music industry's most effective marketing instrument in the future. Engagement data, for example, offer insights into how audiences respond to new music genres, trends, artists and songs. This in turn allows professionals from across the music industry to use the data to attract increased visibility for signed artists, target various audience segments with different advertisements, or track subtle listening patterns for improved business decisions. In addition to this, engagement data can also be used to improve recommendation engines by tracking the time spent watching a music video, so that videos with long watch times appear first in search queues.

Marin (2018) continues by accounting for an important aspect from the business perspective of customer segmentation. Since fan activity and consumption patterns vary by different factors such as genre and engagement, marketing strategies will also have to differ by a record label's artist or genre for them to reach their full potential. Music labels will thus have to identify how to separately target these unique customer segments. Ali & Siddiqui (2017) state that a challenge for this genre-based segmentation approach lies in today's large music databases and their maintenance, since the warehouses require exhausting and time-consuming work, particularly when categorizing song genres manually. In addition to this, music can also be derived into many genres and various subgenres, not only on the basis on the music itself, but also on the lyrics. Moreover, the definition of music genres has been

proven to evolve and dynamically change rather quickly and significantly over time. A way to solve this problem would be a type of automatized genre recognition system.

As a response to the demand of handling large quantities of various musical data, Müller (2015) explains in his preface that since the beginning of the 2000's, music processing and music information retrieval (MIR) have developed into a vibrant and multidisciplinary area of research. The field brings together several fields including information science, audio engineering, computer science and musicology. Lerch (2012, 2) provides a general description about the field of MIR. The main emphasis in MIR lies in information extraction through digital signal processing. However, MIR may also include the retrieval and analysis of information that is music-related, but cannot be directly extracted from the audio signal, such as the lyrics, user ratings, performance instructions in the score, or bibliographical information such as the name of the publisher, the publishing date or the work's title.

Rosner & Kostek (2018) point out that automatic musical genre classification (AMGC) is one of the most popular domains in the field of MIR. A general description of AMGC is provided by Chathuranga & Jayaratne (2013). AMGC is the process of classifying music into genres by a machine. The tasks in AMGC mostly consist of the selection of the best features and development of algorithms to perform the classification. A key problem of the topic is to efficiently and effectively extract low-level audio features for high-level classification. Nasridinov & Young-Ho (2014) concur with the previous statement by explaining that the process of AMGC generally consists of two main steps: feature extraction and classification. The first step consists of obtaining audio signal information, while the second step consists of classifying the music into various genres based on the extracted features. The success rate of AMGC can thus be derived into feature extraction and classification.

1.2 Problematization

While the field of MIR is relatively new, Sturm (2013) describes that the GTZAN data set has grown to become the most-used public data set in AMGC-related research. The GTZAN data set was created and first used by Tzanetakis & Cook (2002). Since its introduction in 2002, it has until 2013 appeared in at least 100 published works, which is roughly 40% of all published AMGC-related work. The data set consists of 1000 half-minute song excerpts, which span across 10 equally distributed genres, and low-level features that have been extracted from the excerpts.

Although its high utilization rate, the data set has received wide criticism. Sturm (2013) criticizes the data set for containing exact repetitions, recording repetitions, artist repetitions and version repetitions. Exact repetitions, meaning that the time-frequency fingerprints are highly similar, have been found in 50 excerpts. Recording repetitions, meaning that multiple excerpts have been drawn from the same song, have been found in 21 excerpts. Artist repetitions, meaning that multiple excerpts have been drawn from the same artist, have been found to occur in nearly all excerpts, since most of the excerpts within a genre consist of approximately 10 different artists. For the reggae genre, more than one third of the excerpts have been derived from Bob Marley's music. Finally, version repetition, meaning that excerpts have been drawn from multiple versions of the same song, such as from a remix, a live version or a cover, have been found in 13 excerpts.

In addition to containing different forms of repetitions on a rather large scale, the data set has according to Sturm (2013) also been criticized for its lack of metadata, and from the fact that approximately 10% of the excerpts contain mislabeled genres, while approximately 2% of the excerpts contain clipping and distortion. Moreover, the absence of subgenres in the GTZAN data is verifiable, as it only includes the genres blues, classical, country, disco, hip hop, jazz, metal, pop, reggae and rock. Flexer (2006) shows that a similar lack of subgenres can be perceived in the second most used data set, ISMIR2004, which uses even broader genre definitions for its six genres: classical, electronic, jazz/blues, metal/punk, pop/rock, and world.

All the previously presented factors are contributors to both overoptimistic classification results and to an uncertainty regarding a large proportion of previous AMGC-related research. Panagakis et. al (2009) explain that previous research shows varying mean classification accuracy rates for the task of AMGC. Using the GTZAN data set, the mean classification accuracy has ranged from approximately 61% to 82%, while Sturm (2012) addresses that mean classification accuracies as high as 90% have been reported.

In addition to the previously presented factors, that contribute to overoptimistic classification accuracies, Müller et al. (2011, 18) address that AMGC-related research often seems to have reached a point beyond which it has become very difficult to make any improvements. One possible improvement tactic would be to restrict the domain by explicitly focusing on a limited subset of music genre (e.g. building a beat tracker that is specialized for jazz, and another that is specialized for classical music).

1.3 Objective

Given the extent of uncertainty in previous research, the objective of this study is to provide a reliable re-evaluation concerning the application of machine learning for the task of AMGC. Firstly, the integrity problems occurring in at least 40% in previous research up until 2013 will be completely accounted for. Secondly, the domain is going to be restricted to include only subgenres of heavy metal music, since it was proposed as an improvement tactic by Müller et al. (2011, 18).

Heavy metal was chosen as a restricted domain, since previous research on the topic has to my knowledge only been conducted as theses by Tsatsishvili (2011) and Mulder (2014). Moreover, the fuzziness of the subgenres in heavy metal music could be considered as a reliable way of re-evaluating the true potential of applying machine learning for the task of AMGC. From a business perspective, the objective may be of interest for streaming services, record labels and marketing companies within the music industry. Moreover, the objective is to provide the concepts intuitively through a comprehensible pipeline, without laying too much focus on the underlying mathematics.

The research questions trying to be answered are:

1. Can machine learning be used to distinguish between subgenres of heavy metal music?

2. What is the most effective learning algorithm for distinguishing between subgenres of heavy metal music?"

3. What insights can be derived from the outcome of the process?

1.4 Method

To answer the first research question, low-level features that closely resemble the ones originally used by Tzanetakis & Cook (2002) are going to be extracted from a total of 500 song audio files. The audio files consist of five equally distributed subgenres from a personal collection of MP3-files, so that each subgenre is represented by 100 song audio files. The subgenres were carefully chosen to represent the whole spectrum of metal music in general. The chosen subgenres were (traditional) heavy metal, thrash metal, death metal, black metal and folk metal. Each song within a subgenre has a minimum length of three minutes, but no specified maximum length.

To account for all the integrity problems (i.e. exact repetitions, recording repetitions, artist repetitions and version repetitions) in previous research, it was chosen that the unique artist and song amount were to be 500. To account for the problems of clipping, distortion or other anomalies, the quality of each song file is manually checked by the writer. Finally, to account for the problem of mislabeled genres, each song is manually labelled by the writer, who has approximately 14 years of listening experience regarding various subgenres of heavy metal music.

Applied machine learning models for AMGC are built and evaluated using a wide range of learning algorithms. Due to the balanced classes, classification accuracy in conjunction with confusion matrices were chosen as the primary model evaluation methods. To make the results comparable in future research, results from the final models are also reported using precision, recall and F_1 metrics. Depending on the results from the final models, a conclusion should be reached regarding whether the subgenres could be distinguished from each other or not.

For the second research question, the results from the models are weighted against each other by their classification accuracy. Six of probably the most popular machine learning algorithms for a classification problem were chosen to be used. These are Naïve Bayes, K-Nearest Neighbors, Decision Trees, Support Vector Machines, Random Forests and AdaBoost. In addition to these, an artificial neural network is included, which falls under the subfield of deep learning. This makes the total amount of learning algorithms to be compared seven. To reach a conclusion on the second research question, the models are ultimately ranked based on the classification accuracy from the final models.

For the third research question, some insights are trying to be derived from the outcome of the process. As an example, it may be of interest to distinguish between what parameters were of importance during the feature extraction, what parameters were of importance during the classification, what genres were most often mixed with each other and if some songs were more prone to be misclassified than others.

The programming language Python, the audio analysis package Librosa, the machine learning package Scikit-learn, and the deep learning package Keras with TensorFlow as backend are the primary tools used for the technical part of the research.

1.5 Previous research

In previous work, Tsatsishvili (2011) used a custom created data set with 833 one-minute excerpts across 17 different subgenres of heavy metal, so that each subgenre was represented by 49 excerpts. Any reasonable results were not achieved. In response to this, a subset of the original data set was extracted, so that the total amount of subgenres was narrowed down to seven. The more distinctive subgenres were black metal, death metal, melodic death metal, gothic metal, heavy metal, power metal and progressive metal. In this data set, each subgenre consisted of 30 excerpts with a one track per artist strategy. The extracted features were heavily influenced by the ones used in the GTZAN data set. K-Nearest Neighbors, AdaBoost and a custom-made algorithm inspired by Barbedo & Lopes (2007) were used as learning algorithms, which at greatest achieved classification accuracies of 37.1%, 45.7% and 44.8%, respectively.

Mulder (2014) used a custom created data set with 17 different subgenres, in which the excerpts for each genre were extracted from approximately five full albums from different artists. K-Nearest Neighbor classifiers and a custom-made classifier based on the Mahalanobis distance were used as learning algorithms. Two feature sets, named horizontal interval feature and vertical interval feature, were separately used as input data. Despite the integrity problems with artist repetitions in the data set, classification accuracies between 25% to 30% were achieved with the K-Nearest Neighbors algorithm. The custom-made classifier achieved a classification accuracy between 18-20%, which is still greater than guessing by chance.

1.6 Disposition

This study is divided into six chapters. The chapters are constructed on the already previously cited statement by Nasridinov & Young-Ho (2014), that AMGC generally consists of two main tasks: feature extraction and classification. Thus, Chapter 2 explains how audio signals can be represented in various domains, how features are extracted from audio signals, and how the extracted features are aggregated for further analysis. Finally, the general sonic differences between the five subgenres chosen for this study are explained.

The beginning of chapter 3 introduces the main learning types in machine learning. It will then narrow down on the pipeline of building a machine learning model for a classification problem. Starting from how the extracted data should be explored and preprocessed, the study will move on to explain the tasks of model selection and model validation. Furthermore, the

important concepts of the bias-variance tradeoff and the curse of dimensionality are introduced. After this, suitable performance metrics for evaluating the models are introduced. The chapter ends with an intuitive explanation of the seven learning algorithms to be used.

The empirical study is explained in detail in chapter 4. It binds together previously explained information from the literature review in chapters 2-3 by presenting the full creation process of the applied models. In addition to this, results from the final models are visualized through tables and graphs. Chapter 5 is reserved for result discussion. A Swedish summary is provided in chapter 6.

2. Feature Extraction and Music Information Retrieval

2.1 Music Representation Methods

From a technical point of view, music can be represented in three ways: sheet music, symbolic, and audio, as explained by Müller (2015, 1). Firstly, sheet music stands for visual representations of a score. The original medium of this representation is paper, although it is now also accessible on screens through digital images. Secondly, a symbolic representation of music refers to any machine-readable data format that explicitly represents musical entities. An example of a symbolic representation would be the MIDI-format, where timed note events represent pitches, velocities, and other parameters to generate the intended sounds. Finally, audio representations such as WAV or MP3-files do not explicitly specify musical events. Instead, these files contain coded representations of acoustic sound waves, which are generated when a source creates a sound that is transmitted to the human ear as air pressure oscillations. An audio representation is also the most realistic way of representing music.

Müller (2015, 19) further explains that in contrast to the two other music representation methods, an audio representation encodes all information needed to reproduce an acoustic realization of a piece of music. This includes components such as the temporal, dynamic and tonal micro deviations that make up the specific performance style of a musician. Even though audio representations are the most realistic method of music representation, a drawback of them exists from an AMGC perspective. The cause for the drawback is that the note parameters of audio representations (e.g. onset times, pitches and note durations) are not given explicitly. Instead, all the components result in a single representation of sound, known as an audio signal. Due to the inability of effectively differ note parameters from each other, the analysis and comparison of audio signals are considered challenging tasks for a machine. From here on, the focus will explicitly lie on audio representations, since this is the approach that was chosen for this study.

2.2 Audio Signals and Waveforms

Müller (2015, 19) explains how sound waves of audio representations are perceived by machines in comparison to humans. When a sound is produced, the alternating pressure travels through the air as a wave. This wave can then be perceived as sound by the human, or alternatively converted into an electrical signal by a microphone and thus perceived by a machine. Graphically, the change in air pressure at a certain location can be represented as oscillations (i.e. waves) by a pressure-time plot, also referred to as the waveform of the sound.

The waveform shows the deviation of the air pressure from the average air pressure. The waveform is also said to occur in the time domain, since it depicts the whole signal at once.

Sine waves or sinusoids are the simplest type of waveforms. They are stationary, continuous and of infinite length, as stated by Davy (2006, 21). The characteristics of a sine wave are explained by Müller (2015, 21). The sine wave is completely specified by its frequency (i.e. the reciprocal of the time (measured in seconds) that is required to complete a cycle of an oscillation), its amplitude (i.e. the peak deviation of the sinusoid from its mean), and its phase (i.e. determining where in its cycle the sinusoid is at time zero). The frequency is expressed in Hertz (Hz) and it determines the pitch of the sine wave, where a higher frequency means shorter oscillations and thus, a higher pitch. The sound resulting from a sine wave is called a pure tone or harmonic sound, as it can be considered the prototype of an acoustic realization of a musical note.

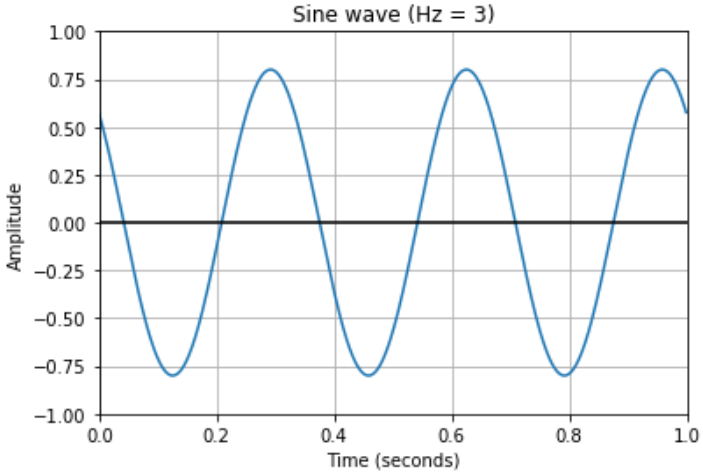

Figure 1. A low-pitched sine wave represented in the time domain as a waveform. This sine wave takes a third of a second to complete an oscillation, making its reciprocal 3/1 = 3 Hz.

In comparison to sine waves, audio signals in music are exceedingly convoluted. Klapuri (2006, 5) explains that musical audio signals may be divided into monophonic signals and polyphonic signals. In monophonic signals, only one note is sounding at a time. When several notes are sounding at the same time, overlapping happens and the signal is called complex or polyphonic. Musical scores are inevitably complex signals. Serizel et al. (2017, 1) clarify that it is most of the time nearly impossible to identify or localize sound events from a waveform, unless they occur at a dynamic range (such as a loud noise in a quiet environment). This is exemplified in Figure 2A, which illustrates distorted guitars, bass and drums kicking in after

an approximately two-second long orchestral intro. The distinction is clearer during the application of harmonic-percussive source separation (HPSS), in which the percussive component of the signal is approximately decomposed from its harmonic component with transparency, as shown in Figure 2B.

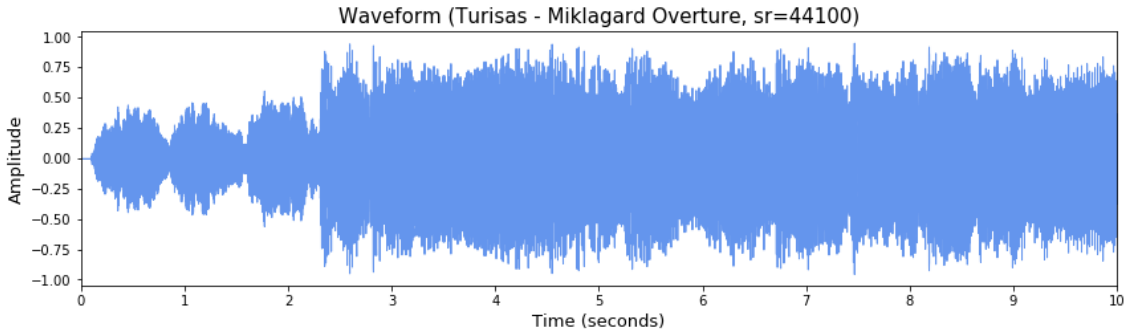

Figure 2A. A complex audio signal represented in the time domain as a waveform. First ten seconds from Miklagard Overture by Turisas.

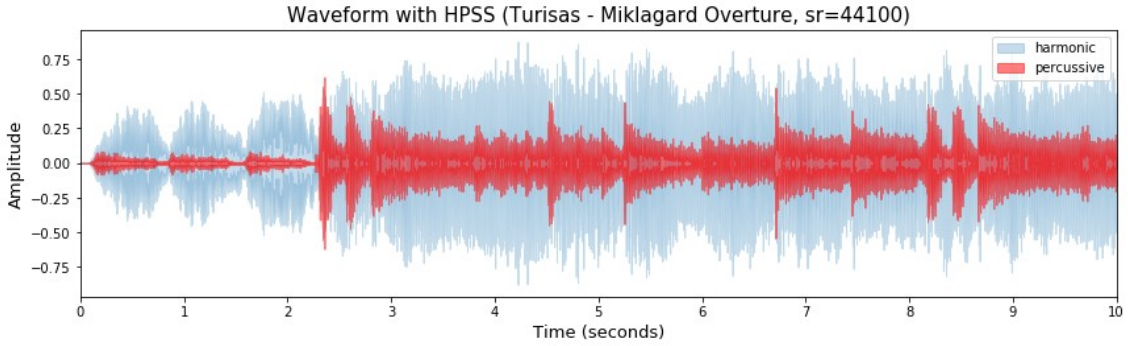

Figure 2B. The same audio signal as in Figure 2A, but with the harmonic and percussive components decomposed from each other.

Another essential aspect to consider is that since a signal is an infinite and uncountable amount of numbers, it needs to be discretized in a digital domain. Lerch (2012, 9) explains that the discretization in time at specific times is known as sampling. According to the Nyquist-Shannon sampling theorem, the sampling frequency must be greater than twice the frequency one wishes to produce. Since the human perception of sound is roughly 20 Hz to 20 000 Hz, an ideal sampling rate would be at least 40 000 Hz.

2.3 Short-Time Fourier Transform and Spectrograms

It has been established that both sine waves and complex audio signals can be visualized in the time domain through waveforms. In contrast to sine waves however, the content of complex audio signals is nearly impossible to meaningfully interpret from waveforms, as shown when comparing Figure 1 with Figures 2A and 2B. The challenge of interpreting meaningful information from waveforms is not only true for humans, but also for machines. Scaringella et al. (2006, 4) explain that waveforms of complex audio signals cannot be directly applied for analysis, since the information they contain are so dense and low-level.

In order to enable feature extraction, Serizel et al. (2017, 1) explain that audio signals are prior to any analysis generally converted from the time domain to either the frequency domain or the time-frequency domain. Müller et al. (2011, 3) establish that the most popular tool used for this transformation is the Short-Time Fourier Transform (STFT). The essentiality of the STFT is emphasized by McFee et al. (2015, 23) in the official Librosa document, which clarifies that nearly all calculations in Librosa rely on the STFT.

Without diving deeper into the mathematical technicalities of the STFT, it can be stated that the STFT is a mathematical formula which, according to Müller et al. (2011, 3), applies the Discrete Fourier Transform (DFT) across partially overlapping short windows of fixed duration, which results in a complex-valued output. For a finite signal, the output of the STFT leads to a description of the time-varying energy across different frequency bands. In other words, the output of the STFT for a finite signal is a sequence of vectors, where each vector is an ordered array of numbers that corresponds to the frequency band differentiated energy of a single frame. For a single frame, the output of the STFT can be visualized in the frequency domain through a spectral envelope, which is also referred to as a spectrum. A snippet from the small orchestral intro for the song Miklagard Overture by Turisas is exemplified as a spectrum in Figure 3 below. The x-axis depicts the frequency bands, while the y-axis depicts the energy amount in decibels (dB).

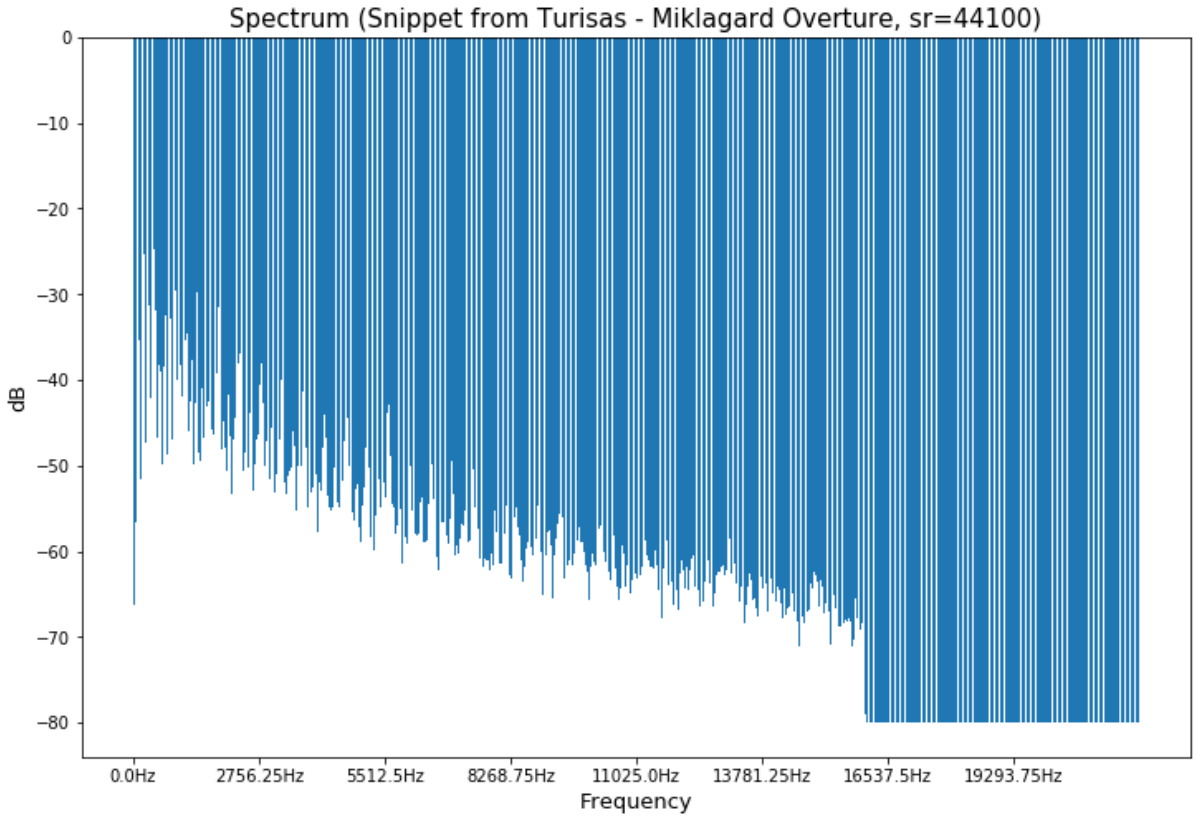

Figure 3. Representation of the spectral energy in the frequency domain. A snippet from the beginning of the song Miklagard Overture by Turisas.

Even though a spectrum could be averaged over multiple frames to represent a finite signal, it is more sensible to apply it for a single point in time. For a finite signal, a more meaningful way of representing the time-varying energy is to use a spectrogram. Lerch (2012, 21) explains that a spectrogram combines the time and frequency components for each (partially overlapping) frame, that was calculated during the STFT. The result is an image in which the magnitude of each frame is represented by a column, and each bin is darkened according to its level of energy. In other words, a spectrogram could be perceived as an ensemble of spectrums, where a narrow vertical slice of the spectrogram corresponds to a single spectrum.

Müller et al. (2011, 3) further elaborate that while a linear spectrogram may be used to represent the STFT, the spectrogram in sound related applications is preferably applied on a logarithmic scale, known as a log-frequency spectrogram. The reason for the suitability lies in the human perception of sound, which follows a logarithmic scale. The logarithmic scale of human hearing means that each doubling in frequency (i.e. an octave) corresponds to an equal musical interval. For example, the lowest note on a piano, A_0, corresponds to 27.5 Hz, while

the octave above it, A_1, corresponds to 55 Hz. A_2, on the other hand, corresponds to 110 Hz and so on.

Gibson et al. (2014) concur with Müller et al. (2011, 3) concerning the logarithmic aspect of human hearing. It is specified that the Mel scale is a specific type of logarithmic scale that approximates the human hearing through the application of overlapping and asymmetric triangular filters over the frequency domain. The motive of this so-called filter bank can most intuitively be understood through an example. For instance, as humans, it is easier to differentiate between sinusoids of 100 Hz and 200 Hz, than it is to differentiate between sinusoids of 1500 Hz and 2000 Hz. Thus, binning of the frequency domain on the Mel scale more accurately represents how a human perceives an audio signal. The output for representing a finite audio signal with the STFT on the Mel scale is known as a Mel-frequency spectrogram or melspectrogram, which is exemplified in Figure 4A. The time-varying energy is illustrated with time on the x-axis and the frequency on the Mel scale (measured in Hz) on the y-axis. The color levels in turn depict the intensity of the spectral energy in dB. Figure 4B shows the same signal with a different sampling rate.

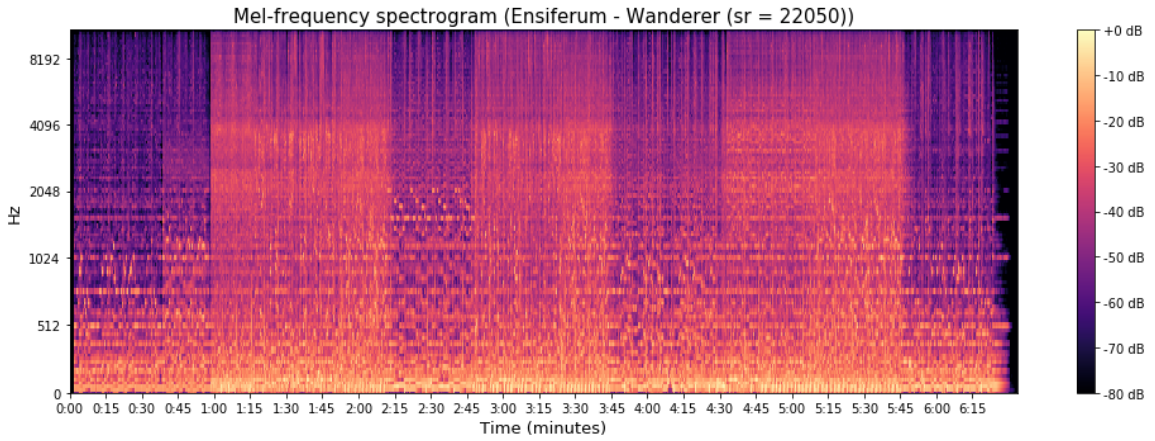

Figure 4A. Wanderer by Ensiferum represented in the time-frequency domain as the Mel-frequency spectrogram. The signal has been downsampled to a sampling rate of 22050 Hz. The three transfers from the acoustic to the non-acoustic sections are clearly distinguishable at approximately 1:00, 2:45 and 4:30.

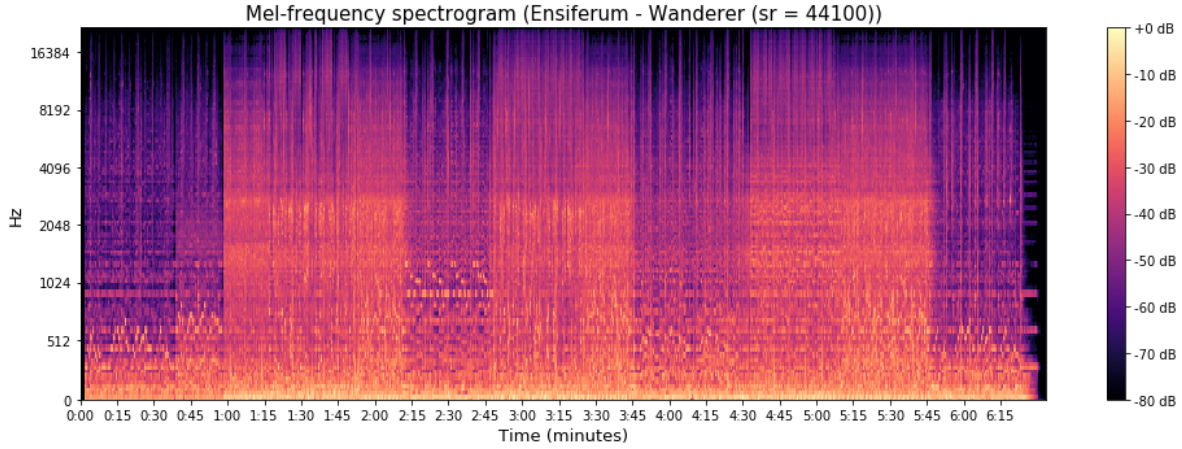

Figure 4B. The same signal as in Figure 4A, but with a sampling rate of 44100 Hz, which is the standard for CD's. When compared to Figure 4A, this spectrogram shows that only a minor portion of the spectral activity is lost, when downsampling to 22050 Hz.

2.4 Low-Level Feature Extraction

In chapter 2.3 it was explained that the transformation of a waveform to its Fourier components through the STFT alters the audio signal from the time domain to the analysis friendly time-frequency domain, which can be visually represented as a spectrogram on different scales. The actual output of the transform is a sequence of feature vectors, where each vector corresponds to a narrow column of the spectrogram. Each feature vector in turn consists of short-term data, that allows the extraction of low-level features.

Lerch (2012, 31) elaborates that the term low-level feature, instantaneous feature, short-term feature or descriptor is generally used for measures that generate one value per short frame of audio samples. Such features are not necessarily meaningful by themselves from a musical, musicological or perceptual perspective, but they can serve as a building block for the construction of high-level features, which describe meaningful properties of the audio signal in a more semantical way, such as pitch, timbre, tempo and loudness.

The following section will provide an explanation on how the low-level features used in this study are extracted on a frame-to-frame basis. As stated earlier, the chosen features bear similarities to the ones used originally in the GTZAN data set by Tzanetakis & Cook (2002).

2.4.1 Mel-frequency Cepstral Coefficients

The Mel-frequency Cepstral Coefficients (MFCC's) are perceptually motivated features, which are, according to Müller et al. (2011, 10), by far the most popular way of describing the spectrum within an individual analysis frame. They have shown to be particularly effective in speech recognition tasks.

Müller et al. (2011, 10) continue by explaining that MFCC's are created by simulating a bank of roughly 40 bandpass filters in the frequency domain, so that the filters are uniformly distributed on the Mel scale. The log-power of the signal is then calculated within each band, and finally the discrete cosine transform (DCT) is applied to the vector of log-powers to obtain the MFCC's, from which typically only 10 to 15 of the lowest coefficients are retained. The properties of the DCT and DFT are briefly the same, except that the coefficients in DCT are real instead of complex-valued, and that the DCT provides a better energy compaction capability, as explained by Shekokar & Mali (2013, 1).

An example of how the MFCC's are derived is shown in Figure 5 below. The figure shows a total of 20 triangular bandpass filters between 0 Hz to 8000 Hz, spread on the Mel scale. Each bandpass filter estimates the spectral energy within the range it falls into.

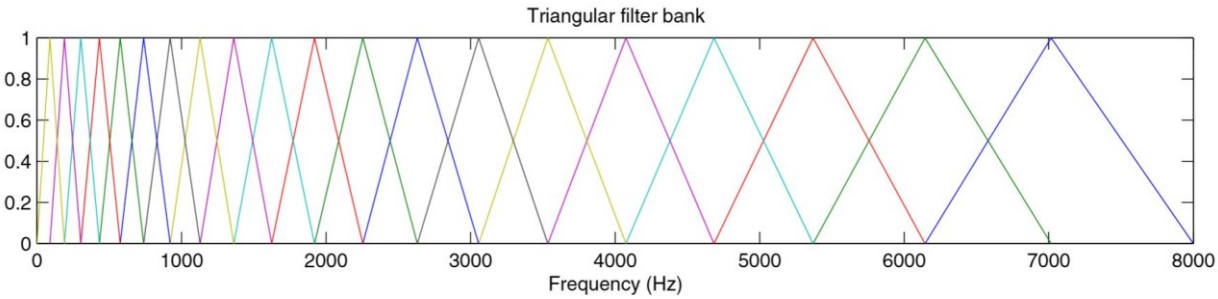

Figure 5. Illustration of the filter banks used in MFCC's for estimating the spectral energy of a spectrum. Source: Qin et al. (2013)

2.4.2 Spectral Centroid

The spectral centroid is a timbral feature, that estimates the brightness of the sound based on the distribution of the spectral energy. According to Tzanetakis & Cook (2002), the spectral centroid is defined as the center of gravity of the magnitude of the spectrum resulting from the STFT. For example, a high centroid value indicates that a large portion of the spectral energy occurs in the higher frequencies.

2.4.3 Spectral Bandwidth

Serizel et. al (2017, 9) explain that the spectral bandwidth is derived from the spectral centroid. The spectral bandwidth resembles the standard deviation of a normal distribution, as it describes the normalized spread of the spectral centroid for each frame.

2.4.4 Spectral Roll-off

The spectral roll-off is another measure of the spectral shape. Tzanetakis & Cook (2002) explain that it is for each frame defined as the frequency below which 85% of the magnitude distribution is concentrated (i.e. the frequency of the 85^{th} percentile of the total spectral energy). The roll-off may also be adjusted to some other percentile in Librosa.

2.4.5 Root-Mean-Square Energy

Serizel et. al (2017, 9) explain that the energy is one of the most straightforward, yet important spectral features. For each frame, the total energy can be computed directly as the sum of the squared amplitude components, resulting in a variable called Root-Mean-Square Energy (RMSE). Intuitively, the RMSE roughly estimates how loud a signal is.

It is clarified on the homepage of Librosa, that RMSE can be either be calculated directly from an audio signal or from a spectrogram. However, using a spectrogram will give a more accurate representation, since its frames can be windowed.

2.4.6 Zero-crossing Rate

In contrast to the previously introduces features, which are all derived from the frequency domain, the Zero-crossing rate (ZCR) is derived from the time domain. Shete & Patil (2014, 2) explain that ZCR is a feature, which is given by the number of times the signal amplitude for a given frame crosses the zero value. Rough estimates about the spectral properties can be obtained using a representation based on the short-time average ZCR.

2.4.7 Dynamic Tempo

Another feature to be derived from the time domain is the dynamic tempo. Despite being time-based at its core, the dynamic tempo utilizes the frequency domain. McFee et al. (2015, 20-21) explain that since the spectral energy at the onsets (i.e. beats) tends to be higher, the peak positions from the onset strength can be used to estimate the onsets. This is achieved by applying an onset detection function over the time-frequency domain to acquire an estimation about the beat locations, similarly to the example shown earlier in Figure 2B. The estimated beats can in turn be used as an estimation of the dynamic tempo.

2.5 Aggregation Methods

Since the output of the extracted features are given on a frame-to-frame basis (e.g. a song that was during the STFT windowed to 4000 analysis frames, will have an output of 4000 values for RMSE), certain aggregation methods are required to represent the features for each song in a single data set.

A simple, yet popular method is to aggregate the results from each frame by its means and variances. Means and variances as aggregation methods have for example been used by Tzanetakis & Cook (2002, 295), while Bahuleyan (2018) used the closely resembling means and standard deviations instead.

2.6 Genre Taxonomy

So far it has been explained how waves of sound can be perceived by both humans and machines, how audio signals can be represented in the time domain, the frequency domain and the time-frequency domain by machines, and how meaningful low-level features can be extracted from audio signals and further aggregated. In contrast to machines, the human

perception and differentiation of music is not mathematical, but subjective. This human perception of music is essential for machines, since the labeling of music genres relies entirely on humans. The following section will explain how humans generally differentiate between music styles.

Chaturanga & Lakshman (2013) explain that music can be divided into many categories based on descriptors such as rhythm, style, mood or cultural background. However, one of the most used descriptors is the musical genre. Musical genres are categorical labels created by human experts used for categorizing, describing and comparing songs, albums or authors in music. Musical genres are often differentiated from each other based on a collection of various descriptors such as pitch, content, instrumentation, rhythmic structure and timbral features. Furthermore, the human perception of music is also dependent on a variety of personal, cultural and emotional aspects, and thus, boundaries among genres may be subjective and fuzzy.

A contributor to the fuzziness of musical genres is the dynamic nature of musical styles. Weinstein (2000, 7) explains that musical styles generally follow a pattern of formation, crystallization and decay. During the formation, the distinction between the new style and the styles out of which erupts are still unclear. Later, in the period of crystallization, the style is self-consciously acknowledged. Its audience recognizes it as a distinctive style, but the boundaries of that style are not rigid. Decay happens, if the style becomes so familiar and predictable that both the composer and the audience start to lose interest in it.

The complexity of music audio signals and the lack of a static and universal genre taxonomy makes the classification process prone to error, especially when distinguishing between musical subgenres. The upcoming part will present the generally recognized sonic characteristics (i.e. instrumentation, song structure, vocal styles and production) of the five subgenres chosen for this study. In addition to this, some historical context will be presented with the goal of providing an oversight of the relationship between the subgenres. It should be mentioned that a large portion of the characteristics are also related to lyrical, cultural and performance aspects. However, only the sonic characteristics are considered, since they directly affect the outcome of the low-level feature extraction, and thus, the outcome of the classification.

2.6.1 Heavy Metal

Heavy metal is a music genre that may itself be classified as a subgenre of rock music. Weinstein (2000, 14-16) explains that the formative phase of heavy metal occurred in the late 1960's and early 1970's. Heavy metal derived its basic song structure, its fundamental chord progressions, and its guitar riffs from blues rock. Moreover, heavy metal added influences from psychedelic music and acid rock to the structure of blues rock.

The fundamental sonic dimensions in heavy metal are explained by Weinstein (2000, 22-25). The essential sonic element in heavy metal is power, expressed as loudness. The guitar is often distorted, and in addition to its rhythmic possibilities, also played as a lead instrument. Guitar solos are essential, while a wide range of electronic gadgetry, such as wah-wah pedals and fuzz boxes are often used to treat sound not merely as notes of discrete duration and pitch, but as tones that can be bent into each other. Moreover, the drum kit is far more elaborate than the drum kits employed for many other forms of rock music, which allows the rhythmic pattern to take on complexity within its elemental drive and insistency. The distinctive bottom sound provided by the bass drum is greatly enhanced by the electronic bass guitar, which performs a more important role in heavy metal than in any other genre of rock music. No other instruments are part of the standard role, although keyboards are permitted.

Another distinguishable sonic characteristic of heavy metal are the vocals and its interactions with the other instruments. Weinstein (2000, 25-26) explains that there is an intimate connection with the vocals and the instruments, with the voice participating as an equal, not as a privileged instrument. A crucial aspect is that the singer's voice must sound very powerful and be emotionally expressive. The range of emotions is often wide, including pain, defiance, anger, and excitement. Some singers may use an operatic voice to achieve this, although seldomly a pure toned one. Special sounds, especially screams, serve to emphasize the power and emotionality of the voice.

Since its emergence, the sound of heavy metal continued developing until it crystallized in the mid-to late 1970's. This period of growth would ultimately function as a basis for a further division of various subgenres of heavy metal, as explained by Weinstein (2000, 21).

2.6.2 Thrash Metal

The formation of thrash metal is explained by Weinstein (2000, 48-49). The thrash metal subgenre, which is also sometimes referred to as speed metal, evolved in the beginning of the

1980's during the golden age of heavy metal. Thrash metal was most directly influenced by a specific segment of British heavy metal, which came to be known as the New Wave of British Heavy Metal. Moreover, thrash metal also bears the trace of the punk explosion, which took place in England during the mid-1970's.

Thrash metal bears sonic similarities with heavy metal. A general distinction between heavy metal and thrash metal is provided by Weinstein (2000, 48-49). In contrast to heavy metal, the dominant distinction of thrash metal is an increase in tempo. Furthermore, thrash metal also often lacks the arty, heroic and overblown song structures, which tend to be present in heavy metal. Pillsbury (2006, 5) concurs on the previously explained by stating that the primary musical difference lies in the consistent treatment of tempo in a rhythmically intense manner and as a distinctly aggressive musical element.

In addition to an increase in both tempo and overall aggressivity, Pillsbury (2006, 10-11) further explains that a musical aesthetic of thrash metal is to play riffs using a substantial amount of palm muting. Palm muting is a guitar technique that offers a large amount of timbral shading and control based on slight alterations of the guitarist's picking hand during performance. This percussive guitar technique allows timbral variation, as it emphasizes both the lower frequencies and the very high overtones of the sound envelope, as well as cutting out the mid-range. The combination of speed and timbre allows the production of the so called "mosh parts" of thrash metal songs. These energetic sections can for example either cut sixteenth-note intensities into eight-note intensities or maintain a sense of continuous rhythm in the guitars, while being accompanied by a half-time drum pattern.

2.6.3 Death Metal

While thrash metal has its roots in heavy metal, Kahn-Harris (2007, 3) and Weinstein (2000, 51) concur that death metal has primarily emerged from thrash metal. Kahn-Harris (2007, 103) elaborates that American bands played a central role in the formation of death metal in the mid-1980's, and that the crystallization happened in the late 1980's to early 1990's in the USA.

Purcell (2003, 11) provides a definition about the most distinguishable sonic characteristics of death metal. Death metal is generally referred to as any music characterized by a combination of down-tuned instruments, fast drumming (and the use of the so called "blast beat" technique), churning riffs, and gruff vocals that can be screamed, but are typically grunted in

a low guttural voice. In general, death metal is most easily distinguishable for its vocals. A vast majority of death metal bands uses very low, beast-like, almost indiscernible growls as vocals. Many bands also use high and screechy vocals, or simply deep and forcefully sung vocals.

Another central characteristic is the technical nature of the music, which is explained by Purcell (2003, 12-13). Death metal is extremely different in comparison to popular music, predominantly because it is more complicated, more difficult to play and technically impressive. The time changes and scale patterns can be numerous, while traditional song frameworks are often ignored. This may result in complex song structures, resembling the song structures often heard in classical or jazz music.

Kahn-Harris (2007, 106) continues by elaborating that a second formation phase happened in the early 1990's in Sweden when a number of bands started to play a highly distinctive form of death metal featuring noticeably melodic rhythm patterns. The music is considered so distinctive, that it is sometimes treated as a separate subgenre, known as melodic death metal. Bowar (2017) concurs that melodic elements play a major part in the core sound of melodic death metal. Harmonic guitar work, acoustic guitars and keyboards are prevalent, while death metal style growls are mixed with harsh screams and tuneful harmonies of clean singing. In this study, both death metal and melodic death metal songs are included in the data set under the subgenre of death metal.

2.6.4 Black Metal

In parallel with death metal, the black metal subgenre also began its formation phase in the 1980's, largely inspired by both thrash metal and death metal. Kahn-Harris (2007, 4-5) explains that the crystallization of the subgenre happened in the early 1990's. Norwegian bands had predominantly developed a highly distinct and influential form of black metal, characterized by screamed, high-pitched vocals, extremely rapid tempos, tremolo riffs, a trebly guitar sound, and simple production values. Olsen (2008, 9) continues by stating that while virtually all earlier forms of metal had emphasized clarity, energy and virtuosity, black metal music is dense, deeply distorted and cacophonous. Black metal changes the guitar solos, technical wizardry and song structure of traditional metal for a buzzing, droning wall-of-sound.

Hagen (2011, 184-187) provides a detailed explanation about how the sonic characteristics in black metal differ from the ones in thrash metal and death metal. Black metal eschews the propulsive palm muted riff-drive of thrash metal and death metal in favor of a swirling and distinct atmosphere. An essential element for achieving this atmosphere are the tremolo riffs, a technique in which single strings are picked at extremely fast tempos. However, unlike in thrash metal and death metal, the guitars often sound thin and brittle. Keyboards may thus be used to de-emphasize the centrality of the guitars. As an alternative to the low growls in death metal and the aggressive barks in thrash metal, the vocals in black metal are high-pitched screams. Also, in contrast to thrash metal and death metal, the blast beat as a drumming technique is not used to create rhythmic propulsion, but instead used to operate in a hypermetric relationship with the chord progressions. As a result of these factors, the overall sonic dimension is more centered on harshness and timbral density, than on low frequencies.

2.6.5 Folk Metal

Folk metal, which according to Weinstein (2014, 59-60) is sometimes also interchangeably referred to as pagan metal or viking metal, developed by incorporating folk elements into metal music. Marjenin (2014, 37) describes the uncertainty surrounding its formation phase. While some say that the formation began with the integration of folk elements into heavy metal music in Europe during the early 1990's, others argue that folk metal developed from black metal. The sole predecessors of folk metal are however not only limited to earlier subgenres of metal, as folk music and folk rock has also influenced the development of the subgenre. Weinstein (2014, 62) annotates that the prominence of folk metal exploded in the early 2000's, particularly in Finland.

Marjenin (2014, 53-54) continues by explaining the most prominent sonic characteristics of folk metal. The most distinguishable characteristic of folk metal is the presence of folk instruments in a heavy metal context. Folk instruments provide an opportunity to increase the soundscape produced by the typical heavy metal instrumentation. Despite the capabilities of keyboards being able to simulate the sound of any instrument, many bands still choose to incorporate the original, acoustic instruments into their ensemble. The chosen folk instruments may vary, and they might reflect the band's place of origin. For example, bands from Ireland might choose to include a fiddle, bodhrán or uillean pipes. The potential assortment of pre-modern instruments used by folk metal bands are however wide. Weinstein

(2014, 65) exemplifies that folk metal bands may include a violin, tin whistle, mouth harp, kantele, accordion or various horns in their ensemble.

The incorporation of folk tunes is another distinguishing sonic characteristic of the folk metal subgenre. Marjenin (2014, 55) explains that folk tunes might be quoted in heavy metal compositions, while some songs might be entirely built around a folk melody. For example, a main instrumental played on the guitar may reference a folk melody or it may be a folk melody performed in its entirety, while other parts of the song such (e.g. the verse or a middle eight) may follow Western chord progressions.

3. Classification and Machine Learning

3.1 Learning types in Machine Learning

At the end of chapter 2 it was explained how humans classify music and what uncertainties are tied to the task. Chapter 3 explains how a machine can learn the task of genre classification from the data that is fed to it.

The way a machine learns can be divided into multiple learning types. There is no consensus on how many learning types exist, but Murphy (2012, 2) addresses that the learning types in machine learning are generally divided into two main types: supervised learning and unsupervised learning. A further commonly occurring division is portrayed by Heidenreich (2018), who divides the learning types into supervised learning, unsupervised learning and reinforcement learning. The suitability of the learning type depends on the available data and on the problem to be solved. The following part will explain the three main learning types in machine learning.

3.1.1 Supervised Learning

Murphy (2012, 2) explains that the goal in supervised learning (or predictive learning) is to learn a mapping from inputs x to outputs y, given a labeled set of input-output pairs. In the simplest setting, each training input is a numeric vector. As an example, the vector can represent different sonic characteristics of a song through a collection of aggregated low-level features. A feature can also be a more complex structured object such as an image in the form of a spectrogram. Marsland (2015, 6) clarifies that based on the training data, the algorithm should through generalization be able to predict sensible outputs for inputs that weren't encountered during the learning. Ideally, the algorithm should be able to deal with noise, which is small inaccuracies in the data that are inherent in measuring any real-world process.

As with input variables, Murphy (2012, 2) explains that the output variable, which is interchangeably referred to as the target variable or the response variable, can also be in any form. The target variable can exist on a real-valued scale (e.g. the predicted amount of internationally sold physical records for June can be somewhere between 0 and the total amount of pressed records). The target variable can also occur as a categorical (or nominal variable) from some finite set. As an example, the predicted genre can be limited to exactly five categorical values (e.g. "black metal", "death metal", "folk metal", "heavy metal", "thrash metal"). If the target variable is real-valued, the problem is called regression, while if

the target variable is categorical, the problem is called classification. Thus, the task of AMGC is a classification problem.

3.1.2 Unsupervised Learning

A second learning type is known as unsupervised learning (or descriptive learning). It differs from supervised learning, since no target variable is provided in the input data, which can be true in real world applications. Murphy (2012, 2) explains that the lack of a target variable means that the goal in unsupervised learning is much less well-defined, since neither a right nor wrong answer exists. This also means that unsupervised learning lacks any real error metrics. Unsupervised learning is therefore often used to simply find interesting patterns in the data.

Bishop (2006, 3) exemplifies on the use cases of unsupervised learning by explaining that it can be applied in pattern recognition problems. The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called clustering. In addition to this, unsupervised learning can be used to determine the distribution of data within the input space, which is known as density estimation. Finally, the projection of data from a high-dimensional space down to two or three dimensions for the purpose of visualization can also be performed through unsupervised learning algorithms.

3.1.3 Reinforcement Learning

A third learning type is known as reinforcement learning. Kober et al. (2013, 1) explains that reinforcement learning offers a framework for robotics, by letting a so-called learner autonomously discover an optimal behavior through trial-and-error interactions with its environment.

Alpaydin (2010, 447) explains further that the learner, which is also known as the agent, receives a reward (or penalty) for its actions in trying to solve a problem in the environment. After a set of trial-and-error runs, the agent should learn the best policy, which is the sequence of actions that maximizes the total reward. Lampropoulos & Tsihrintzis (2015, 32) clarify that reinforcement learning differs from supervised learning because neither the input nor the output pairs are presented. Instead, the agents fall into a state at a specific time, where they select an action and therefore get a consequence.

3.2 Model Building

Chapter 3.1 explained the three most common learning types in machine learning. The following part explains a general pipeline for the steps needed to build an applicable model for supervised learning.

3.2.1 Feature Engineering

The first step in building a model is obviously to collect the raw data through some method, such as gathering a collection of song audio files. After the data has been collected, the first step to model building is generally the successful extraction of features. In machine learning terminology, feature extraction is often also referred to as feature engineering. The importance of feature engineering is explained by Domingos (2012, 82-83). The most important factor regarding the success of a model are the features used in training. Having many independent features that each correlate well with the output variable raises the probability for successful learning.

Zheng & Casari (2018) explain that feature engineering is the act of extracting features from raw data and transforming them into formats that are suitable for machine learning modeling. It is a crucial step in the machine learning pipeline, because the right features can ease the difficulty of modeling, and therefore enable the output of higher quality results. Since data and models are universally so diverse, it is difficult to generalize the practice for feature engineering for different types of problems and data sets. Sarkar et al. (2018, 183) concur on the previously explained by stating that feature engineering is both an art and a science to transform data into features for feeding into models, and that successful feature engineering requires a combination of domain knowledge, experience, intuition and mathematical transformations.

3.2.2 Data Exploration and Preprocessing

After the features have been extracted, one needs to have an understanding about what the extracted features look like, so that they can be properly preprocessed for further analysis. This step is known as data exploration. The data might for example be checked for any missing values to see if some error occurred during feature extraction. Various data visualization and summarization methods may also be used to gain a visual understanding of the data.

After an understanding of the data has been achieved through data exploration methods, it is often revealed that the data needs to be preprocessed in some way, so that it can be compiled into a structure, that allows it to be properly and reliably used for further purposes. This process is referred to as data preprocessing, data cleaning or data preparation.

Zhang et al. (2003) provide examples of data preprocessing methods. Data preprocessing can happen by imputing missing values to treat missing data, by eliminating duplicate records, and by removing anomalies or outliers for consistency. Kotsiantis et al. (2006) fill in on the previously explained, by elaborating on the importance of discretization and normalization. Discretization reduces the number of possible values of a continuous feature. This happens by partitioning the possible values of a feature into bins, since a continuous feature with a large amount of possible values could contribute to slow and ineffective learning. Normalization, or scaling, is the process of scaling down the features, so that all features obtain the same weight in a model, regardless of their original scale. This is necessary, since the scales between different features may differ significantly. Min-max normalization and z-score normalization are commonly used normalization methods.

Kotsiantis et al. (2006) continue by underlining the importance of data preprocessing. The preprocessing of data can often have a significant impact on the generalization performance of a model, since the success of the models are dependent on the quality of the input data. The presence of irrelevant, redundant, noisy or unreliable data often leads to difficulties in knowledge discovery during the training phase. Zhang et al. (2003) continue by adding that since real-world data may often be incomplete, inconsistent and noisy, it has been found that data preprocessing generally takes approximately 80% of the total data engineering effort and is thus very time consuming.

Finally, the critical topic of noise in data is addressed in more detail by Alpaydin (2010, 30-31). Noise is any unwanted anomaly in the data, which may affect the learning process in an unwanted way even in a simple hypothesis class. Three different types of noise are presented. Firstly, there may be imprecisions in recording the input features, which could lead to imprecise input data. Secondly, there may be errors in labeling the instances in data. This is also known as teacher noise, since it is a consequence of human behavior. Finally, noise may consist of additional hidden features, which have not been accounted for in the model at all. These hidden features may affect the classification process and are thus modeled as a random component and included as noise.

The complexity of polyphonic audio signals within the fuzzy domain of heavy metal subgenres would for example indicate, that the extracted features contain a significant amount of noise in the form of hidden features, which could negatively affect the learning process.

3.2.3 Model Selection and Parameter Tuning

The preprocessed data should now be ready for the next phase, which is known as model selection. Shalev-Shwartz & Ben-David (2014, 144) explain that model selection is the process of choosing a learning algorithm and its optimal parameters for a chosen task. Interchangeably, this is also known as choosing the class of the model and its hyperparameters, as stated by VanderPlas (2017, 348). The type, amount and tunability of the parameters are dependent on the learning algorithms, which will be introduced in chapter 3.4.

Model selection is an iterative task and is thus considered to be where the heart of machine learning lies at. In practice, model selection happens by trying to solve a given problem for a chosen amount of iterations, so that the parameters are slightly altered for each iteration. In the end, a conclusion about the optimal parameters should be reached. Bennett & Parrado-Hernández (2006, 1266) provide a more mathematical explanation by stating that a model is typically trained to solve an optimization problem that optimizes the parameters of the model with respect to a selected loss function and possibly some regularization function. However, as Murphy (2012, 24) explains, no universally best model exists, which indicates that a set of assumptions that works well in one domain may work poorly in another. The lack of a universally best model is also known as the “no free lunch theorem”.

3.2.4 Model Validation and Resampling Methods

The next process in model building is to validate the selected model. The purpose of model validation is to ensure that that the parameters were properly optimized during the model selection phase, as explained by VanderPlas (2017, 35). Intuitively, the process is also iterative, since each iteration from the previous phase of model selection should also be validated through some method.

The goal in model validation is to reliably estimate the model’s ability to generalize from previously unseen instances. Hastie et al. (2009, 219) state that the results from model validation estimates the model error rate (i.e. the true generalization performance of the model), since it relates to its prediction capability on independent test data. Model validation is extremely important, since it guides the model selection and provides a measurement of the

quality of the ultimately chosen model. Several methods for model validation exist and the three most common methods will be presented next.

3.2.4.1 Validation Set Approach

The simplest way to validate a model is known as the validation set approach, and is explained by James et al. (2013, 176). The validation set approach is done by randomly splitting the available input data into two sets: a training set and a validation set (or test set) during the model selection phase. Using a mathematical learning algorithm, the model is first trained (or fitted) to predict the output variable of the instances in the training set. After the training phase, the trained model is used to predict the target variable of the unseen instances in the validation set. The percentage of the correctly classified instances from the validation set (i.e. the classification accuracy) could then be used as an evaluation metric for the true generalization performance of the model. Several other evaluation metrics exist, and they are later introduced in chapter 3.3.

The simplicity of the validation set approach comes with two drawbacks, which are explained by James et al. (2013, 178). Firstly, the outcome can be highly dependent on the split itself (i.e. which instances were included in the training and the validation set). Secondly, a big portion of the training data is lost, when splitting the data into the training and validation set (the split is typically 60/40). Since statistical methods tend to perform worse when trained on fewer instances, this would put some unnecessary restriction on the learning.

As a response to the two drawbacks in the validation set approach, James et al. (2013, 175) elaborates on resampling methods as solutions to the problems involved. Resampling methods are statistical tools involving repeatedly drawing samples from a training set and retraining a model of interest on each sample to obtain additional information about the trained model. Two of the most common resampling methods are known as cross-validation and the bootstrap, which will be presented next.

3.2.4.2 K-Fold Cross-Validation

Perhaps the most common, yet effective validation method is known as K-Fold Cross-Validation (K-Fold CV). James et al. (2013, 181-182) explain that this approach involves randomly splitting the set of instances into k groups, or folds, of approximately equal size. A typical choice for k is 5 or 10. The first fold is treated as a validation set, and the model is trained on the remaining folds. The model is trained and tested in a similar way as in the

validation set approach, however the phase is repeated k times (i.e. until every fold has acted as a validation set). This described process results in k estimates of the test error, and the K-Fold CV estimate is computed by averaging these values.

Class imbalances are important to account for, when dealing with K-Fold CV. A method for accounting for class imbalances is to include stratification during the splitting of the data. Alpaydin (2010, 487) explains that stratification is the process of making sure that the classes are represented in the right proportions for each fold, so that the training process is not affected by class prior probabilities. For example, if the genre "folk metal" occurs as a target label in 20% of the instances in the input data, then the splitting should be performed in such a way that the class is also represented in 20% of the instances across all the folds.

Alpaydin (2010, 487) continues by elaborating that the data set can in practice be divided into the total amount of instances in the data set (i.e. a data set with n=500 instances would be split into k=500 folds). The training phase is repeated n times, so that each instance alone functions as a validation set. This process is known as Leave-One-Out-Cross-Validation (LOOCV). However, LOOCV as a validation method can be computationally expensive.

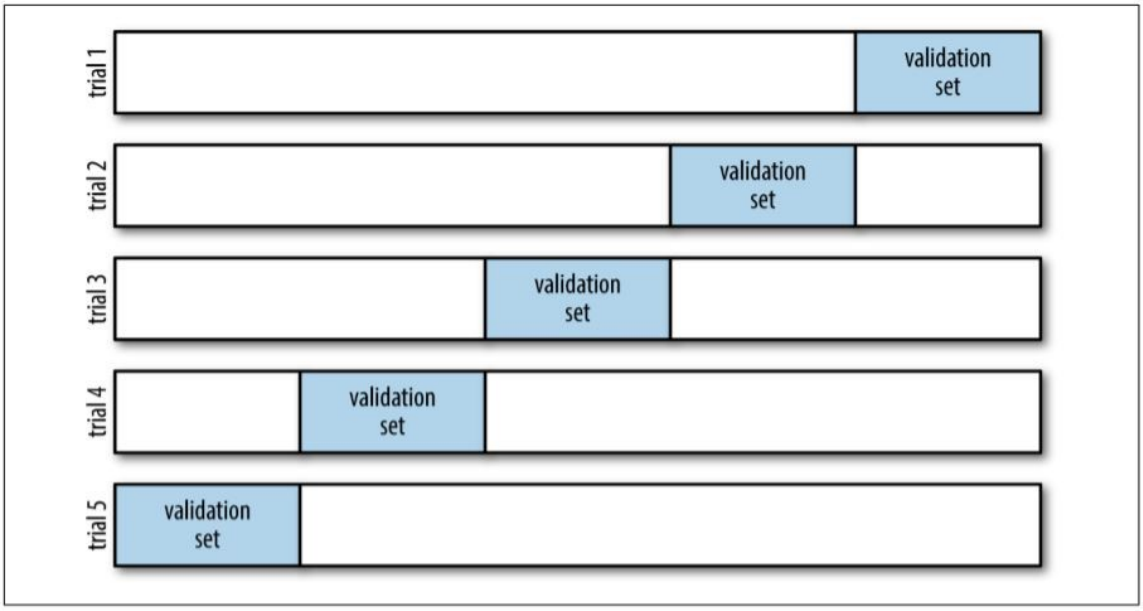

Figure 6. An example of K-Fold Cross-Validation on a data set, in which k = 5. Source: VanderPlas (2017, 362).

3.2.4.3 Bootstrapping

An alternative to the K-Fold CV and LOOCV is bootstrapping, which is explained by Hastie et al. (2009, 249). Bootstrapping happens by randomly drawing instances with replacement from the training set. This means that as each instance is drawn and added to the bootstrap sample, the instance is also added back to the training set and can therefore be redrawn into the same bootstrap sample. To complete a single bootstrap sample, the process of drawing instances should be repeated until it contains as many instances as the original training set.

In practice, a set of bootstrap samples are usually created. Hastie et al. (2009, 249-250) explains that for each bootstrap, the model is fit on the bootstrap. The model is then validated on the original training set. Using the original training set as a validation set could be perceived as controversial, as instances may be overlapping between the two sets. The overlapping instances may contribute to unrealistically good prediction results, which means that bootstrapping does not generally provide valuable estimation of the model error rate in comparison to K-Fold CV or LOOCV.

3.2.5 The Bias-Variance Tradeoff

During the phases of model selection and model validation, there is another important factor to consider than just the choice of the learning algorithm, its parameters and the validation method. A tradeoff known as the bias-variance tradeoff puts directly some restriction on the process of model selection.

An understanding of the two reasons for why a model may perform poorly is required to understand the bias-variance tradeoff. VanderPlas (2017, 364-365) explains that in practice, a model with high bias and low variance does not have enough flexibility to suitably account for all the features in the data. Such a model is said to underfit the data, and thus perform badly, when tested on both the training data and the validation data. In contrast to underfitting, a model with low bias and high variance has enough flexibility to nearly perfectly account for all the features and noise properties in the training data. A model with such high flexibility often performs accurately, if evaluated on the training set, but badly when evaluated on unseen instances in the validation set. Such a model is said to overfit the data. Fundamentally, the model selection phase also includes finding a sweet spot in this tradeoff.

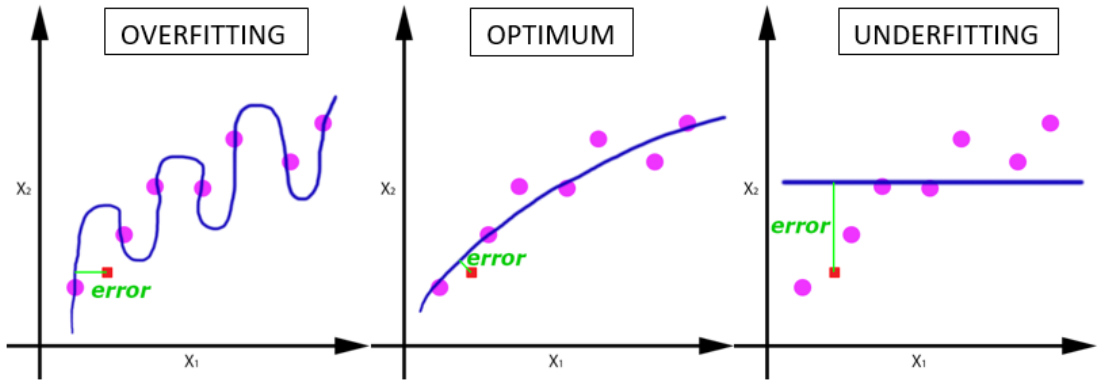

Figure 7. The bias-variance tradeoff illustrated through three regression models. The purple dots depict instances from the training set, while the red squares depict instances from the validation set. On the left: A model with high variance, that overfits. On the right: A model with high bias, that underfits. Source: Sharma (2017).

3.2.6 The Curse of Dimensionality

In addition to the bias-variance tradeoff, another problem to consider in the model selection phase is the curse of dimensionality. Marsland (2015, 17-18) summarizes that as the number of input dimensions (i.e. the number of features) increases, the amount of data needed for the model to sufficiently generalize also increases. This is referred to as the curse of dimensionality. The cause for the curse of dimensionality is, according to Hastie et al. (2009, 22-23), that a greater fraction of the data volume needs to be captured when the amount of dimension grows, which can lead to failure in learning.

3.2.6.1 Feature Selection

Kotsiantis et al. (2007) explain that the curse of dimensionality can be accounted for with feature selection, which is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and may allow learning to operate faster and more effectively.

Sarkar et al. (2018, 242) explain that feature selection is generally divided into three methods. Firstly, filter methods are independent of the inductive learning algorithm. Instead, they use univariate statistics such as correlation for the selection process. Secondly, wrapper methods use recursive approaches to build multiple models using a subset of features, which allows the selection of the best model. Thirdly, embedded methods combine the methods of both filtering and wrapping by leveraging machine learning models themselves to rank features based on relative feature importance. Examples of models that include built-in estimators of

feature importance are tree-based methods like random forests and extremely randomized trees.

3.2.6.2 Principal Component Analysis

In contrast to feature selection, the curse of dimensionality can also be accounted for by scaling down the features through unsupervised learning methods. A popular method for this is principal component analysis (PCA), which transforms the data into a new set of dimensions. Harrington (2012, 274) explains that in PCA, the data is first centered by subtracting the mean feature value for each instance. An eigenvalue analysis is then performed on the covariance matrix of the features, which yields the eigenvectors and their corresponding eigenvalues. The eigenvectors can then be ranked by their eigenvalues and the linear transformation will happen by multiplying the instances with the top N largest eigenvectors.

A more intuitive explanation is provided by Harrington (2012, 271-273). For each standardized feature pair, a line that maximizes the variance is drawn. A linear transformation is applied so that the line with the highest variance becomes the x-axis. The y-axis is in turn attained by drawing the line, that maximizes the variance, while at the same time being orthogonal to the x-axis. Finally, the reduced data is transformed to the new dimensions through multiplication.

3.3 Performance Metrics for Model Evaluation

After the tasks of model selection and model validation have been completed, the model error rate (i.e. the true generalization performance of the model) can be evaluated through various quantitative performance metrics.

3.3.1 Classification Accuracy

The most obvious performance metric would be the classification accuracy. Marsland (2015, 23) mentions that the classification accuracy is simply calculated by dividing the number of correctly classified instances with the total number of instances in the data set.

This simple method has a drawback, since class imbalances can lead to skewed results. For example, consider a binary classification problem where 90% of the instances in the training set are labelled as death metal, while only 10% are labelled as black metal. A model that for some reason classifies all instances as death metal would have an admirable classification accuracy of 90%, even though it has classified all instances of black metal incorrectly.

Furthermore, if more than two output labels exist, the classification accuracy does not provide information on how the instances were mislabeled. As a result of this, more adequate performance metrics are preferred in conjunction with the classification accuracy.

3.3.2 Confusion Matrix

A complementary evaluation method for the classification accuracy is the confusion matrix. Harrington (2012, 143) explains that the confusion matrix shows which classes were confused with each other. The x-axis of the confusion matrix is often chosen to show the actual output label, while the y-axis shows what the model predicted. If all the off-diagonal elements of a confusion matrix were to be zero, a perfect classifier with a classification accuracy of 100% is achieved.

In its simplest case, the confusion matrix occurs in a binary classification problem as a 2x2 matrix. Figure 7 below illustrates a binary classification problem with the possible outcomes "1" and "-1". As an example, a true positive (TP) would indicate that an instance was correctly predicted as "1", while a false negative (FN) would indicate that an instance was incorrectly predicted as "-1". Likewise, a true negative (TN) stands for a correctly predicted "-1", while a false positive (FP) stands for an incorrectly predicted "1".

		Predicted	
		+1	-1
Actual	+1	True Positive (TP)	False Negative (FN)
	-1	False Positive (FP)	True Negative (TN)

Figure 8. A confusion matrix for a binary classification problem. The possible outputs for the two categorical output labels ("1" and "-1") are represented in statistical terms. Source: Harrington (2012, 144).

3.3.3 Precision, Recall, F_1

Even though the confusion matrix allows the identification of which target labels were mixed with each other, it still ignores any class imbalances. However, there are at least three other metrics, that account for class imbalances and that can be directly derived from the values inside a confusion matrix.

While the classification accuracy can be derived by dividing the sum of TP and TN with the total number of instances in the data set, Marsland (2015, 23) provides an explanation of how the other three commonly used metrics, known as precision, recall and F_1 are derived from a confusion matrix. Firstly, precision is for a given class calculated by dividing the number of correctly predicted instances with the total number of predicted instances for that given class. Secondly, recall is for a given class calculated by dividing the amount of correctly predicted instances with the total number of actual instances for that given class. Finally, the metric known as F_1 takes both the precision and recall into account. It can be derived from the formula shown in Figure 9 below.

$$\text{Accuracy} = \frac{\#TP + \#TN}{\#TP + \#FP + \#TN + \#FN}$$

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP}$$

$$\text{Recall} = \frac{\#TP}{\#TP + \#FN}$$

$$F_1 = 2\,\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Figure 9. Formulas for how the four numeric performance metrics can be derived from a confusion matrix. Source: Marsland (2015, 23).

3.4 Learning algorithms

The following section provides an intuitive description about learning algorithms chosen for this study, in their order of interpretability. According to Plate (2000), the interpretability of the statistical models varies according to the trade-off between flexibility and interpretability of the learning algorithm. A model with high flexibility is a more complex one. Such models are more likely to find important relationships in the data and overfit, but also harder to interpret than models with low flexibility. Models with a low flexibility may miss out on some relationships and can thus be prone to underfitting. Such models are however often easier to interpret.

The models are presented in a subjectively approximated order of flexibility, starting from the model with the lowest flexibility and the highest interpretability. For visualization purposes,

the learning algorithms are presented on a two-dimensional plane, which means that only two features are used. As the actual amount of input dimensions is equal to the number of chosen features, the input dimensions in real life applications are often multidimensional and could be represented with a n-dimensional manifold.

3.4.1 Naïve Bayes

Naïve Bayes is probably one of the most robust, but practical learning algorithms. Richert & Coelho (2013, 118-119) explain that, at its core, Naïve Bayes classification is nothing more than keeping track of which feature gives evidence to which class. In Naïve Bayes, all features must be independent of each other for the algorithm to work optimally. This is rarely the case in real-world applications, and hence the name "naive". Nevertheless, Naïve Bayes has a reputation of providing good prediction accuracies in practice even when the independent assumption does not hold.

The general functionality of Naïve Bayes models is explained by VanderPlas (2017, 382). Naïve Bayes models are a group of extremely fast and simple classification algorithms that are often suitable for very high-dimensional data sets. They are built upon Bayesian classification methods, and only have a few tunable parameters. Bayesian classification relies on Bayes's theorem, which is an equation describing the relationship of conditional probabilities of statistical quantities. The goal in Bayesian classification is to find the probability for each output label, given some observed features. This model is called a generative model, because it specifies the hypothetical random process that generates the data. Specifying a generative model for each target label in the training data is the main task in training a Bayesian classifier, which could be perceived as a difficult task. However, making naive assumptions about the generative model for each output label enables a rough approximation of the generative model to be formed, which enables the classification to be performed.

Further elaboration on different methods of specifying a generative model is done by VanderPlas (2017, 383-385). The simplest model can be achieved through Gaussian Naïve Bayes, in which the classifier is built by finding the mean and standard deviation of the points within each output label with no covariance between the dimensions. For each output label, an ellipse is created. An ellipse represents the Gaussian generative model for an output label, with the center of the ellipse illustrating a higher probability. This generative model is used to compute the likelihood of an output label for any data point in the data set.

In addition to the Gaussian Naïve Bayes, Jain (2017) briefly discusses two other commonly used types of Naïve Bayes methods. While the Gaussian Naïve Bayes assumes the distribution of the features to be normal, the Multinomial Naïve Bayes algorithm is used when the data is distributed multinomially, that is, when multiple occurrences matter a lot. Moreover, the Bernoulli algorithm is used when the features in the data set are binary.

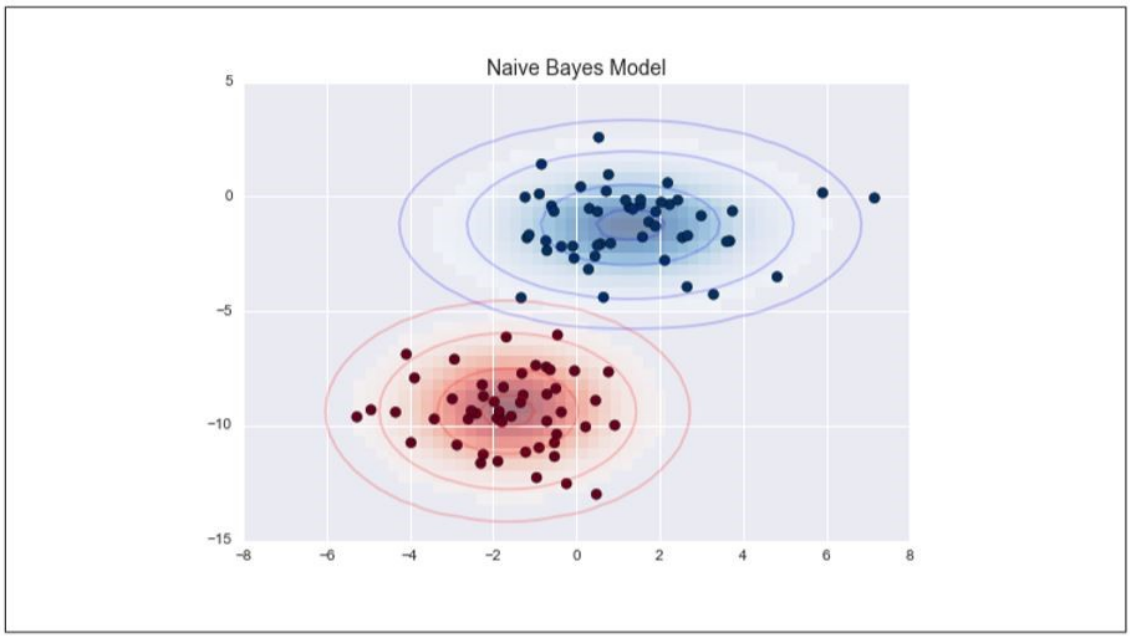

Figure 10. A Gaussian Naïve Bayes model. The ellipses represent the probabilities for the two target labels in a two-dimensional classification problem. Source: VanderPlas (2017, 384)

3.4.2 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a simple probabilistic classifier, which can be surprisingly close to the optimal Bayes classifier. James et al. (2013, 39-40) explain that the goal of the KNN classifier is to estimate the conditional distribution of the output labels given a set of features, and then classify a given instance to the class with the highest estimated probability. In KNN, the number of K is the main adjustable parameter. Setting K = 3 would mean that an instance, x_0, is predicted by looking at its three closest neighbors, and through a majority vote predict the target label of x_0. The KNN approach could also be thought of as a decision boundary, which assigns an output label to all possible coordinates based on the current neighbors of a given coordinate. As K grows, the bias grows, which leads to underfitting. On the other hand, setting K = 1 would indicate that the decision boundary is overly flexible and thus prone to high variance. This would instead lead to overfitting.

Marsland (2015, 158) explains further on the functionality of KNN. For each instance in the training data, the KNN algorithm looks at similar data and decides whether to be or not to be in the same class as them. This requires computing the distance to each instance in the training set, which is computationally relatively expensive, especially in high-dimensional data sets. The choice of K is also not trivial, because a small K makes the data sensitive to noise, while a large K reduces the accuracy, as the observations too far away are considered.

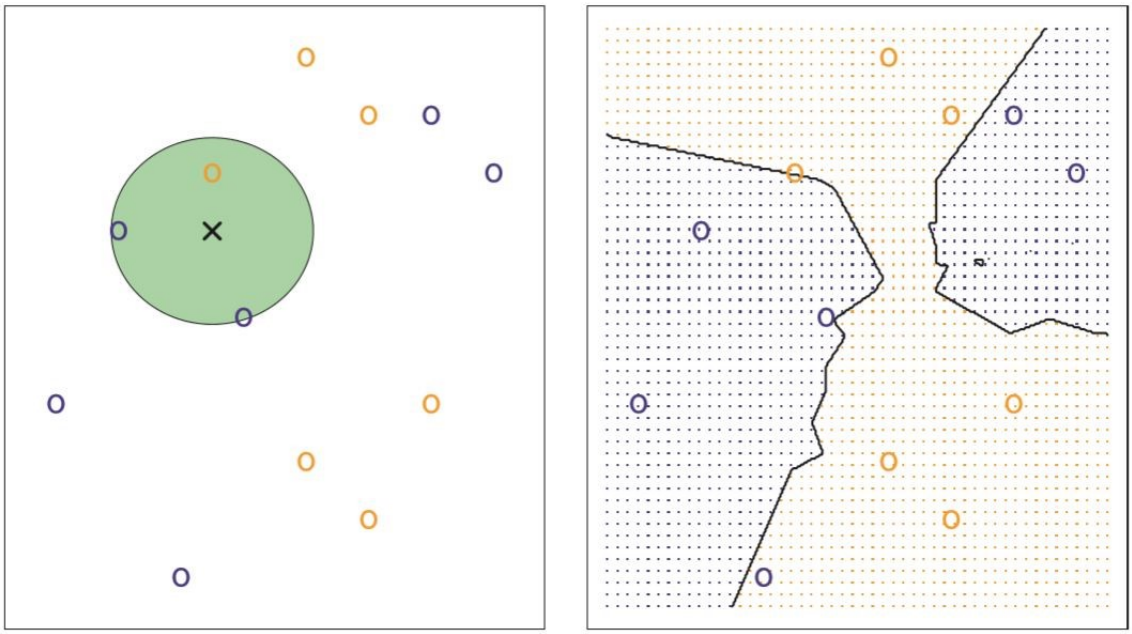

Figure 11. Left: KNN with two target labels in a two-dimensional space, where K = 3. From the majority of its three closest neighbors, the test instance (i.e. the black cross) is going to be predicted as a blue circle. Right: The same example illustrated with decision boundaries. The figure shows which target label is assigned for any coordinate, when K = 3. Source: James et al. (2013, 40).

3.4.3 Decision Trees

A decision tree is a learning algorithm that classifies instances by sorting them based on their feature values. Criminisi et al. (2012, 6-7) explain that decision trees are built upon split nodes, branches and leaves, which are organized in a hierarchical fashion. These are also commonly referred to as internal nodes, edges and terminal nodes, respectively. Each tree starts with a split node. The first split node is also referred to as the root node and it is a function to be applied on the feature that best divides the instances in the data set. VanderPlas (2017, 421-422) continues by explaining that each split node consists of a question, which is sequentially asked to narrow down the options even among a large amount of output labels.

The questions generally take the form of axis-aligned splits in the data, which means that each split node splits the data into two new categories using a cutoff value within one of the features. A split node that splits the data into two categories would indicate that the split node has two outgoing branches, each leading to their own split node.

The criterion of the splitting is defined by a gain measurement algorithm. Boutsinas & Tsekouronas (2004, 174) exemplify that some of the most popular methods for this are the gain, gain ratio, gini and twoing. Shalev-Shwartz & Ben-David (2014, 252) elaborate that among all the possible splits, the algorithm chooses the split that maximizes the gain and performs it, or alternatively chooses not to split the data at all, which indicates that the tree is fully grown and a leaf is reached. The leaf yields the actual prediction of the tree. Figure 12 below illustrates the general tree structure and exemplifies the workflow of a decision tree.

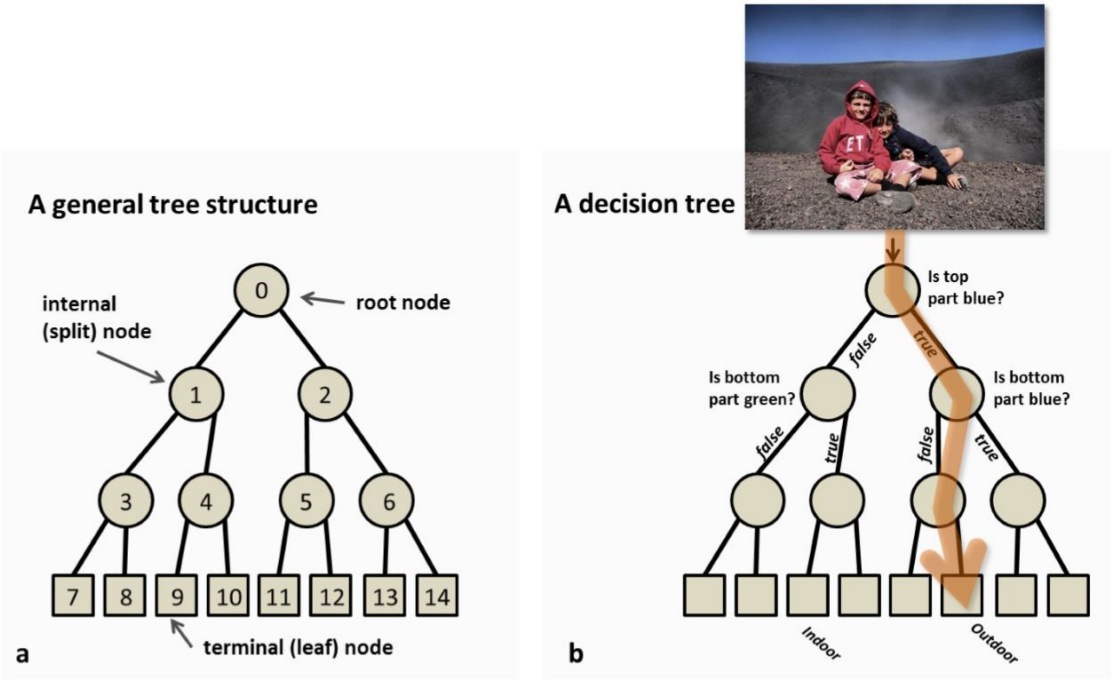

Figure 12. On the left: An illustration of the general structure of a binary decision tree. On the right: An example of the workflow of the same decision tree. Source: Criminisi et al. (2012, 6).

Decision trees may also comprise some issues. Kotsiantis (2007, 252) states that overfitting is a common problem in decision trees and that two common approaches are often used to avoid it. First, the training algorithm can be stopped before it reaches a point at which it perfectly

fits the training data. Secondly, the decision tree can be pruned by various methods, such as applying a threshold on how many nodes should be allowed in the tree.

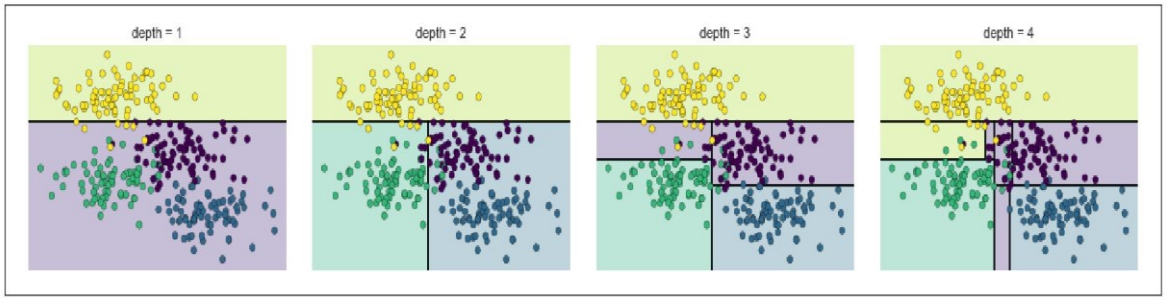

Figure 13. An illustration of how the decision tree iteratively performs four axis-aligned splits in a two-dimensional space for a classification problem with four target labels. Source: VanderPlas (2017, 423).

3.4.4 Support Vector Machines

Support vector machines (SVM's) are a powerful and flexible class of supervised learning algorithms for both classification and regression problems, which are discussed by VanderPlas (2017, 405). In Naïve Bayes classification, a simple model was used to describe the distribution of each underlying output label, and these models were then used to probabilistically determine labels for new observations. On the contrary to Naïve Bayes, the goal of using SVM's in a classification problem is instead to find a line or curve that separates the classes from each other.

An explanation on the general structure of SVM's is provided by Kotsiantis (2007, 260-261). SVM's consist of a separating line, which is known as a hyperplane, and its margins. In theory, multiple variations of hyperplanes could be drawn for them to separate the target labels from each other. The optimal hyperplane is however achieved by minimizing the squared norm of the separating hyperplane. In other words, the hyperplane that creates the largest possible distance between the separating hyperplane and the instances on either side of it is the optimal hyperplane that maximizes the margin. Once this hyperplane has been found, the data points that lie on its margin are known as support vector points and the solution is represented as a linear combination of only these points. SVM's are well suited to deal with learning tasks where the number of features is large with respect to the number of training instances, since the model complexity of support vector machines is unaffected by the number of features encountered in the training data.

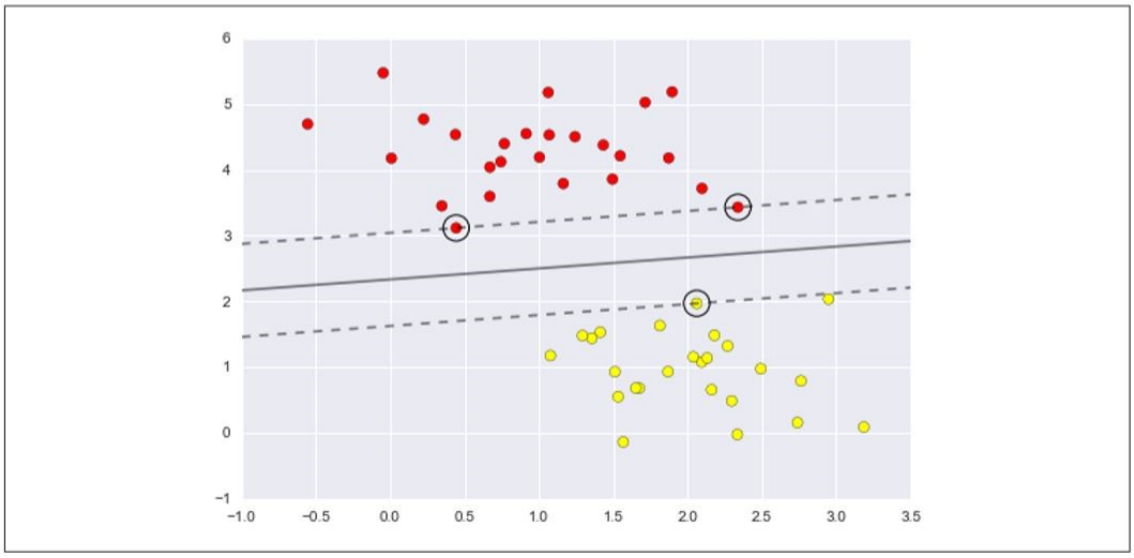

Figure 14. A SVM's classifier in a two-dimensional space with two target labels. The grey line illustrates the margin maximizing hyperplane, while the two dashed lines illustrate its margins. The circled observations are the support vector points. Source: VanderPlas (2017, 409).

3.4.5 Random Forests

Polikar (2012, 1-2) explains that multiple classifier systems, also called ensemble systems, have enjoyed growing attention within machine learning communities during the last couple of decades due to their proved efficiency and versatility. The goal of ensemble systems is to create several classifiers with relatively fixed bias, and then combining their outputs to reduce the variance.

One of the most known ensemble systems is known as random forests, originally developed by Breiman (2001). Polikar (2012, 12) explains that random forests were developed from one of the earliest and simplest ensemble-based algorithms called bagging (short for bootstrap aggregation). Random forests are essentially an ensemble of decision trees trained with a bagging mechanism. Breiman (2001, 2), explains that significant improvements in classification accuracy have resulted from growing an ensemble of decision trees and letting them vote for the most popular class.

An explanation on the functionality of random forests is provided by Shalev-Shwartz & Ben-David (2014, 255-256). As decision trees are prone to overfitting, random forests are used as a way of reducing the danger of overfitting. Each tree in the forest is grown by applying an algorithm on a random subset of instances from the training set. The instances are drawn with

replacements, i.e. bootstrapped. These subsets together form a vector that contains a random, but uniformly distributed subset of instances from the training set. For each subset of instances, the algorithm forms a decision tree. At each splitting stage of the tree, the algorithm is restricted to choose a feature that maximizes the gain. The final prediction of the random forest is obtained by a majority vote over the predictions of the individual trees.

Cutler et al. (2012, 157) provide further details about random forests, by stating that they were developed as a competitor to boosting. They consist of only one or a few tunable parameters, are relatively easy to train, and are suitable for both regression and classification problems of the high-dimensional type. Random forests also contain measures of feature importance, differential class weighting, missing value imputation, outlier detection and visualization.

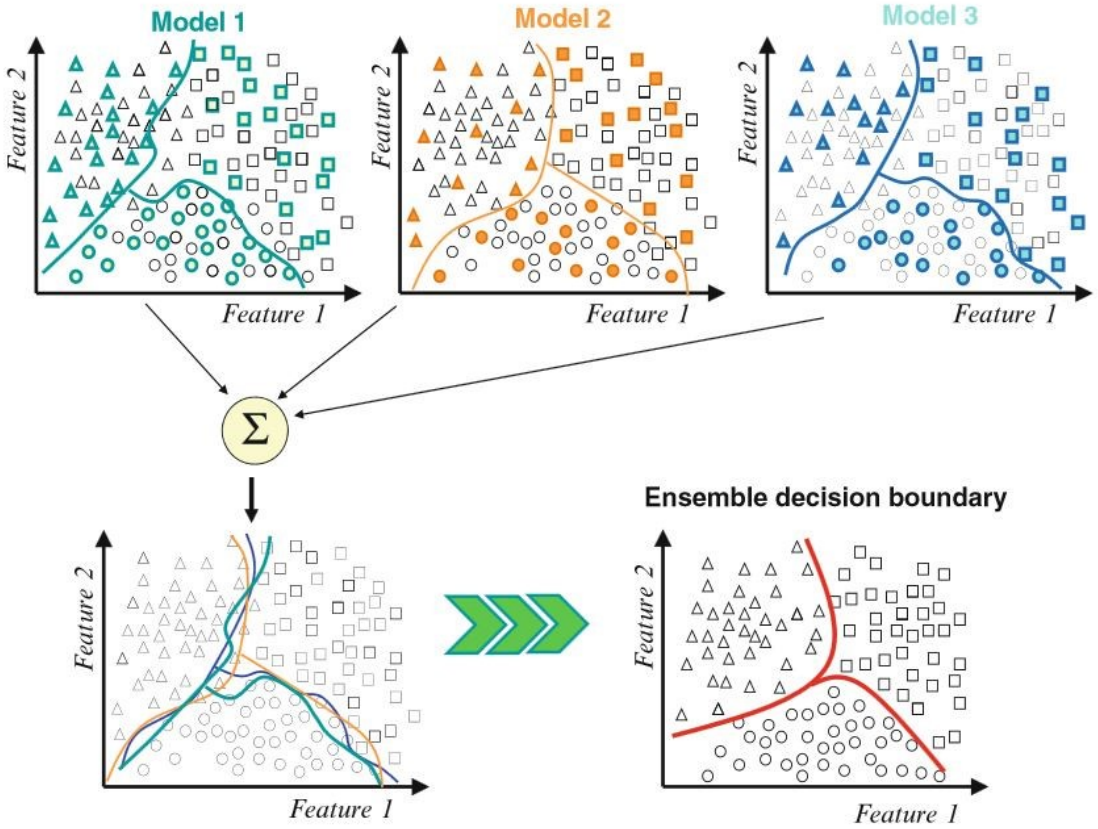

Figure 15. An illustration of a random forest in a two-dimensional space with three target labels. The votes from three decision trees are combined into a single model. For each tree, a colored data point depicts a bootstrapped input vector. Source: Polikar (2012, 3).

3.4.6 AdaBoost

Another general ensemble method for improving the accuracy of any given learning algorithm is called boosting. Schapire (2003, 1-2) explains that building a highly accurate prediction model is difficult, but coming up with very rough rules of thumb, that are only moderately accurate is not. Boosting is based on the observation that finding many rough rules of thumb can be a lot easier than finding a single, highly accurate prediction rule. The boosting approach is implemented by starting with an algorithm that finds these rough rules of thumb. In boosting, these algorithms are named "weak" or "base" learning algorithms.

Boosting has some similarities to random forests, which is revealed by Schapire (2003, 2), who further elaborates on the functionality of a boosting algorithm. Boosting happens by sequentially feeding various subsets of training instances into a base learning algorithm. For each subset of data, the base learning algorithm generates a new weak prediction rule. After several iterations of subsets, the boosting algorithm combines these weak prediction rules into a single prediction rule, which in the end should be much more accurate than any of the single weak rules. Unlike in random forests, Ferreira & Figueiredo (2012, 41) specify that the sampling in boosting does not happen with replacement.

Multiple variants of boosting algorithms exist. Ferreira & Figueiredo (2012, 42) explain that the adaptive boosting algorithm, more commonly known as AdaBoost, is a well-known and deeply studied one, which has shown praiseworthy performance results. The key idea behind AdaBoost is to use weighted versions of the same training data instead of random subsamples of training data. The weak prediction rules with AdaBoost are obtained sequentially, using reweighted versions of the same training data, with the weights depending on the accuracy of the previous weak prediction rules. D'Souza (2018) provides further intuition on AdaBoost, by explaining that the base learners in AdaBoost are often decision trees with a single split node, called decision stumps. When the first decision stump is created by AdaBoost, all instances are weighted equally. To correct the error from the previous stump, the instances that were incorrectly classified now carry more weight than the instances that were correctly classified. The model will continue to adjust the error faced by the previous model, until a sufficiently accurate model is built.

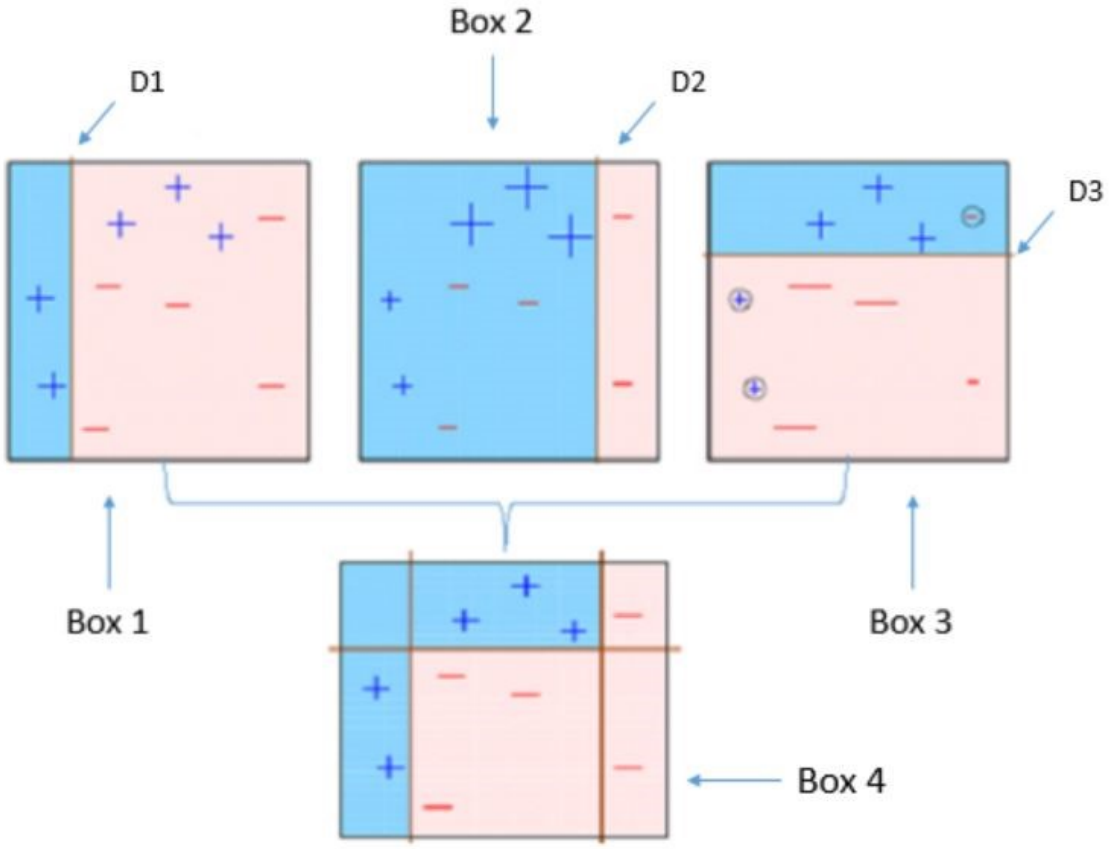

Figure 16. Illustration of AdaBoost in a two-dimensional space with two target labels. After the three misclassified instances during the first decision stump (D1), the algorithm iteratively puts more weight on the misclassified instances in the second decision stump (D2), depicted by the larger plus icons. The three weak prediction rules are ultimately combined into a single prediction rule, shown in Box 4. Source: D'Souza (2018).

3.4.7 Artificial Neural Networks

Artificial neural networks (ANN's), which are sometimes also called multilayer perceptrons, feedforward neural networks or simply just neural networks, are algorithms that are built upon mimicking the biological brain. The main components in an ANN are the neurons and their corresponding weights and activation functions. The neurons are in turn interchangeably also referred to as units.

Brownlee (2016, 38-39) explains that the neurons in an ANN are computational units that have weighted input signals and produce an output signal using an activation function. The amount of weights is dependent on the amount of inputs. As an example, a neuron with three incoming inputs would consist of four weights, one for each input and one for the bias. The activation function of a neuron, on the other hand, is a single mapping of the summed

weighted input to the output of the neuron. The activation function governs the threshold at which the neuron is activated and the strength of the output signal. In other words, the activation function controls to which extent a neuron fires when it is activated.

The neurons themselves are organized in a structure, that makes up a network of neurons. According to Brownlee (2016, 39-41), neurons are arranged into layers of neurons, where the first layer is called the input layer. The input layer takes the input from an actual data set. The layer after the input layer is called a hidden layer, because it is not directly exposed to the input data, but rather to the weighted results of the input layer. An ANN can also consist of multiple hidden layers. The final layer is called the output layer and it is responsible for outputting a value or a vector of values, that correspond to the format required for the problem. In a multiclass classification problem, the number of neurons in the output layer is generally equal to the number of target labels, so that each neuron stands for a probability that a specific target label is assigned for a given instance.

Generally, the actual training process for the ANN happens through an algorithm known as the stochastic gradient descent. Brownlee (2016, 41-42) intuitively explains that a single instance or a batch of instances are fed forward to the network, while the neurons are being activated. When the instance or the given batch of instances reaches the output layer, the expected output is compared to the actual output, and an error is calculated. After this, the so-called backpropagation algorithm is applied. During this step, the error is propagated back through the network, one layer at a time, and the weights are slightly updated according to the amount of error that they contributed to. In other words, the learning is an iterative process where the weights of the neurons are slightly adjusted for each instance or batch that gets fed into the network.

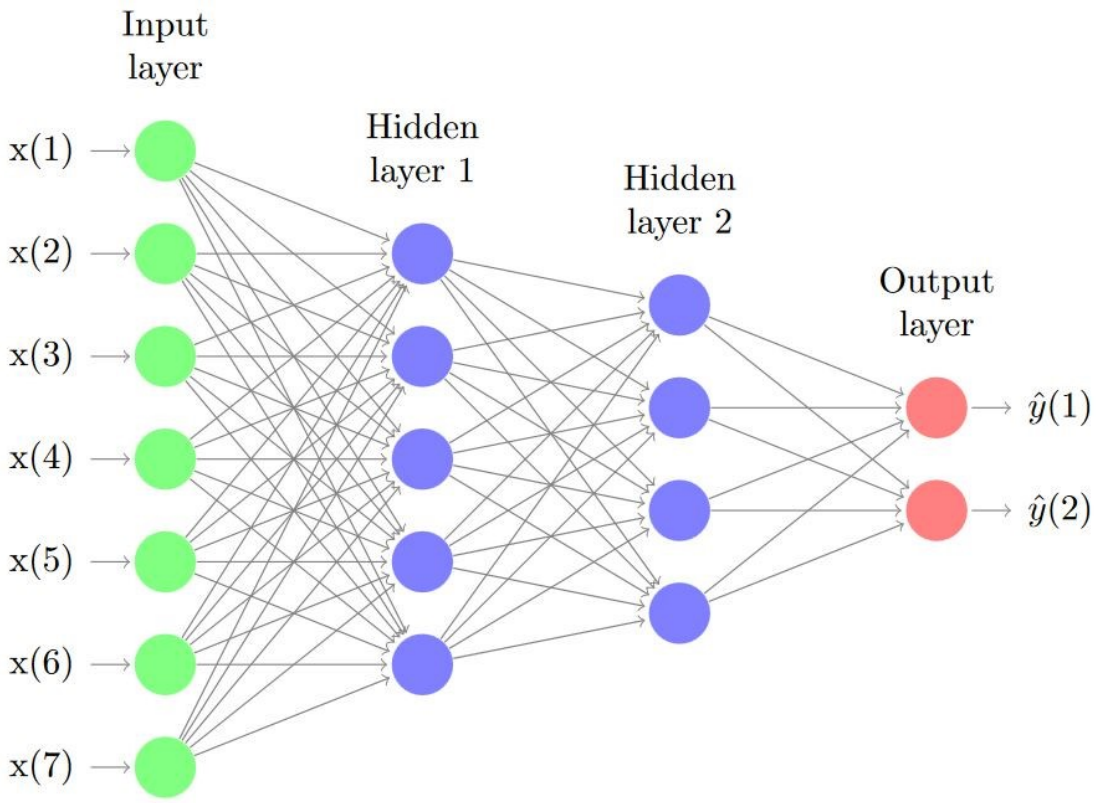

Figure 17. An example of a neural network, which is fed seven features. The network has two hidden layers with 5+4 neurons, and two possible outcomes. Source: Dixon et al. (2016)

ANN's may come in slightly different forms than the general one explained above. Two of the most common ones are convolutional neural networks (CNN's) and recurrent neural networks (RNN's). Brownlee (2016, 116) explains that CNN's are used in image recognition, computer vision and natural language processing problems, since they were developed to differentiate between spatial structures. CNN's are often fed images as inputs. In contrast to CNN's, Brownlee (2016, 170) explains that RNN's are used in a diverse array of problems, such as in language translation and in automatic caption of images and videos, since their use of a so-called long-short term memory allow them to learn and generalize across sequences of inputs, rather than individual patterns.

4. Empirical Study

4.1 Method

The first step of the study is to acquire the optimal parameters for both feature extraction and classification. The methodology is a cyclical research process, which resembles the one originally proposed by Wallace (1969) in Järvinen & Järvinen (2004, 6). Feature extraction is sequentially performed with different sets of feature extraction parameters. For each set of feature extraction parameters, the extracted data is first preprocessed and classification is then performed with a set of test model parameters. The process ultimately leads to the acquisition of the optimal feature extraction parameters based on the results from the three top performing models.

When the optimal feature extraction parameters have been acquired, they are selected for use in the tuning of the model parameters, which in turn will lead to the optimal model parameters. The second step in the empirical study is to report and analyze the results from the optimal models. For reproducibility, the random seed used in this study was set to 42.

4.2 Data Overview

The data set for each iteration in the tuning of the feature extraction parameters has 57 columns and 500 rows, where each row corresponds to the song data for a single song. Columns 1-4 contain text information (i.e. genre, artist name, song name, album name), while columns 5-56 are the aggregated features. The aggregated features were derived by taking the mean and variance of MFCC's 1-20, spectral centroid, spectral bandwidth, spectral roll-off, RMSE, ZCR and dynamic tempo. Finally, column 57 contains information about if any missing values occurred during the feature extraction. This column is used for integrity reasons to check if any anomalies occurred during the feature extraction, so that the given instance could be excluded from the classification process. However, no such instances were found during at any point in the iterations, so all the 500 rows were kept for further processing. Thus, as shown in Table 1 below, the total amount of unique artists, songs and albums for each genre were 100. Table 2 below shows information about the columns of the data sets.

Genre	Artist count	Song count	Album count
Black Metal	100	100	100
Death Metal	100	100	100
Folk Metal	100	100	100
Heavy Metal	100	100	100
Thrash Metal	100	100	100

Table 1. The aggregated unique values for the artist, song and album columns grouped by the genre column after checking the data set for any flawed instances.

#	Column	Description	Type
1.	genre	Genre name	object
2.	artist	Artist name	object
3.	song	Song name	object
4.	album	Album name	object
5.	mfcc1_mean	Aggregated mean, MFCC1	float64
6.	mfcc1_var	Aggregated variance, MFCC1	float64
7.	mfcc2_mean	Aggregated mean, MFCC2	float64
8.	mfcc2_var	Aggregated variance, MFCC2	float64
9.	mfcc3_mean	Aggregated mean, MFCC3	float64
10.	mfcc3_var	Aggregated variance, MFCC3	float64
11.	mfcc4_mean	Aggregated mean, MFCC4	float64
12.	mfcc4_var	Aggregated variance, MFCC4	float64
13.	mfcc5_mean	Aggregated mean, MFCC5	float64
14.	mfcc5_var	Aggregated variance, MFCC5	float64
15.	mfcc6_mean	Aggregated mean, MFCC6	float64
16.	mfcc6_var	Aggregated variance, MFCC6	float64
17.	mfcc7_mean	Aggregated mean, MFCC7	float64
18.	mfcc7_var	Aggregated variance, MFCC7	float64
19.	mfcc8_mean	Aggregated mean, MFCC8	float64
20.	mfcc8_var	Aggregated variance, MFCC8	float64
21.	mfcc9_mean	Aggregated mean, MFCC9	float64
22.	mfcc9_var	Aggregated variance, MFCC9	float64
23.	mfcc10_mean	Aggregated mean, MFCC10	float64
24.	mfcc10_var	Aggregated variance, MFCC10	float64
25.	mfcc11_mean	Aggregated mean, MFCC11	float64
26.	mfcc11_var	Aggregated variance, MFCC11	float64
27.	mfcc12_mean	Aggregated mean, MFCC12	float64
28.	mfcc12_var	Aggregated variance, MFCC12	float64
29.	mfcc13_mean	Aggregated mean, MFCC13	float64
30.	mfcc13_var	Aggregated variance, MFCC13	float64
31.	mfcc14_mean	Aggregated mean, MFCC14	float64

32.	mfcc14_var	Aggregated variance, MFCC14	float64
33.	mfcc15_mean	Aggregated mean, MFCC15	float64
34.	mfcc15_var	Aggregated variance, MFCC15	float64
35.	mfcc16_mean	Aggregated mean, MFCC16	float64
36.	mfcc16_var	Aggregated variance, MFCC16	float64
37.	mfcc17_mean	Aggregated mean, MFCC17	float64
38.	mfcc17_var	Aggregated variance, MFCC17	float64
39.	mfcc18_mean	Aggregated mean, MFCC18	float64
40.	mfcc18_var	Aggregated variance, MFCC18	float64
41.	mfcc19_mean	Aggregated mean, MFCC19	float64
42.	mfcc19_var	Aggregated variance, MFCC19	float64
43.	mfcc20_mean	Aggregated mean, MFCC20	float64
44.	mfcc20_var	Aggregated variance, MFCC20	float64
45.	spectral_centroid_mean	Aggregated mean, spectral centroid	float64
46.	spectral_centroid_var	Aggregated variance, spectral centroid	float64
47.	spectral_bandwidth_mean	Aggregated mean, spectral bandwidth	float64
48.	spectral_bandwidth_var	Aggregated variance, spectral bandwidth	float64
49.	spectral_rolloff_mean	Aggregated mean, spectral roll-off	float64
50.	spectral_rolloff_var	Aggregated variance, spectral roll-off	float64
51.	rmse_mean	Aggregated mean, root-mean-square energy	float64
52.	rmse_var	Aggregated variance, root-mean-square energy	float64
53.	zcr_mean	Aggregated mean, zero-crossing rate	float64
54.	zcr_var	Aggregated variance, zero-crossing rate	float64
55.	tempo_mean	Aggregated mean, dynamic tempo	float64
56.	tempo_var	Aggregated variance, dynamic tempo	float64
57.	missing_values	Total amount of missing values in the pre-aggregated feature arrays	int64

Table 2. An overview of the columns of the raw data set resulting from each feature extraction during the tuning of the feature extraction parameters. The columns numbered 5-56 are the aggregated features.

4.3 Tuning of feature extraction parameters

The sampling rate, the duration and the offset of the signal are tunable parameters, which were specified during the feature extraction. The sampling rate controls the quality of the signal by setting how many samples of the signal are measured per second. The duration is the length of the signal in number of seconds, while the offset allows a given number of seconds at the beginning of a signal to be skipped. It should be noted that if not specified otherwise, the values for the parameters are the default parameters used by Librosa v0.6.2.

As explained in a blog post at the official homepage of Librosa (2019), CD's use a standard sampling rate of 44100 Hz, while digital audio files (e.g. .WAV, .MP3) may have arbitrary sampling rates. Having different native sampling rates across the files would lead to inconsistencies in the feature extraction. Fortunately, sampling rate conversion allows for consistency across all files. Per default, Librosa uses a sampling rate of 22050 Hz, since it reduces memory consumption and decreases the total time of the feature extraction process. Moreover, it is also possible to successfully analyze music without sacrificing much, since the relevant data seems to lie somewhere under a C_9, which corresponds to approximately 8372 Hz. The value of 8372 Hz is well below the 11025 Hz cutoff, which is the resulting cutoff, when using a sampling rate of 22050 Hz. Earlier it was shown in the comparison of Figure 4A and Figure 4B, that only a small portion of the spectral activity was lost with a sampling rate of 22050 Hz. Thus, the sampling rate was first set to 22050 Hz.

To avoid silent parts or non-relevant spectral activity (e.g. artistic intros in the form of environmental or atmospheric sounds) in the beginning of a song to affect the outcome of the feature extraction, an offset of 30 seconds was set. Since the duration of each song file is at least 180 seconds (i.e. 3 minutes), the duration of the signal was set to 150 seconds (i.e. 2.5 minutes). This is the maximum value for keeping a consistent length across all song files, while still enabling the use of the chosen offset time.

The next step was to apply the STFT over the song excerpts. During this step, the window size and the hop length were specified. The window size is the length of audio frames that is considered for each STFT. By default, the window size is 2048. The hop length is the number of frames between each STFT (i.e. how much overlapping should happen between each STFT). A hop length that is equal to the window size would indicate, that no overlapping occurs between the STFT's. A suitable hop length equals to one fourth of the window size

(i.e. 2048/4=512). The window size and hop length were thus permanently set to 2048 and 512, respectively.

After a loaded signal has been transformed to its Fourier components through the STFT, the entirety of the resulting n-dimensional array was transformed into a Mel-frequency spectrogram, while still retaining the original n-dimensional array. The n-dimensional array was used to extract the time domain features, which were the ZCR and the dynamic tempo. In contrast to this, the frequency domain features were extracted from the Mel-frequency spectrogram. These were the 20 MFCC's, the spectral centroid, the spectral bandwidth, the spectral roll-off and the RMSE. Finally, the results were aggregated by their means and variances, which resulted in the aggregated features.

Z-scores were chosen as a normalization method for the aggregated features. Stratified K-Fold CV with 10 folds was chosen as a validation method, which means that each fold consists of 500/10=50 equally distributed observations. Each of the models were trained with test parameters on the training folds and then validated on the test folds. The process was repeated ten times. The classification accuracies were then averaged to achieve a representative mean classification accuracy. Table 3 below shows the average classification accuracies using different sampling rates. Trying higher sampling rates (44100 Hz and 33075 Hz) paradoxically resulted in lower classification accuracies than the default value of 22050 Hz. Surprisingly, further halving the sampling rate to 11025 Hz resulted in even better classification accuracies on average. Going from 11025 Hz in both directions yielded that the optimal sampling rate should be approximately 15000 Hz.

Grid 1: Tuning of the sampling rate (Hz)									
Current Parameters	Song amount = 500, Sampling rate (Hz) = ?, Offset (s) = 30, Song duration (s) = 150, Normalization = Z-scores, n_features = 52, Model parameters = GridSearchCV with test parameters, Validation = StratifiedKFold (n_splits = 10) *Note: * indicates that empty filters in the mel frequency basis were detected due to a low sampling rate.*								
Model / Tuning	**Gaussian Naive Bayes**	**k-NN**	**Decision Tree**	**SVM's**	**Random Forests**	**AdaBoost**	**Neural Networks**	Average (Total)	Average (Top 3)
Sr = 44100 Hz	43.0%	46.2%	38.2%	56.6%	52.6%	44.4%	53.8%	47.8%	**54.3%**
Sr = 33075 Hz	45.2%	45.0%	42.4%	54.2%	50.0%	47.8%	51.2%	48.0%	**51.8%**
Sr = 22050 Hz	53.0%	52.0%	48.0%	59.2%	55.8%	50.4%	54.6%	53.3%	**56.5%**
*** Sr = 19000 Hz**	51.8%	52.8%	51.4%	60.8%	57.4%	51.8%	57.4%	54.8%	**58.3%**
*** Sr = 15000 Hz**	54.4%	53.6%	46.0%	62.8%	60.0%	53.4%	58.0%	55.5%	**60.3%**
***Sr = 11025 Hz**	53.2%	53.6%	44.8%	60.4%	57.6%	54.0%	59.0%	54.7%	**59.0%**
***Sr = 5513 Hz**	50.8%	50.0%	44.8%	58.0%	56.6%	52.4%	56.4%	52.7%	**57.0%**

Table 3. Tuning of the sampling rate.

Since song excerpts of 150 seconds and an offset of 30 seconds were used in the tuning of the sampling rate, the next step was to scrutinize if better classification accuracies could be obtained by performing feature extraction on signals of greater length. Firstly, feature extraction was performed on the full signal (i.e. the duration of the signal length was set to the maximal length and the offset was excluded). Secondly, the maximal signal length with automatic trimming at the beginning and at the end of the signal was used to scale down potential silent part of songs, which could skew the aggregated results of the features. However, this trimming method does not account for non-relevant spectral activity in the form of artistic intros or outros, which could prominently affect the outcome of the classification.

The results can be seen in Table 4 below. Even with the longest track being approximately 13 minutes and the median track length being approximately under 5 minutes, the 150 second song excerpts yielded slightly better results than the maximal length and the trimmed maximal length. This indicates that an excerpt of 150 seconds was enough to generalize the characteristics of a song. As hypothesized, using an offset at the beginning appears to have

had a positive effect on the outcome of the classification, since this method outperformed the two other candidates.

Grid 2: Tuning of the offset (s) and duration (s)									
Current Parameters	Song amount = 500, Sampling rate (Hz) = 15000, Offset (s) = ?, Song duration (s) = ?, Normalization = Z-scores, n_features = 52, Model parameters = GridSearchCV with test parameters, Validation = StratifiedKFold (n_splits = 10)								
Model / Tuning	**Gaussian Naive Bayes**	**k-NN**	**Decision Tree**	**SVM's**	**Random Forests**	**AdaBoost**	**Neural Networks**	Average (Total)	Average (Top 3)
Offset = 30s, Duration = 150s	54.4%	53.6%	46.0%	62.8%	60.0%	53.4%	58.0%	55.5%	**60.3%**
Offset = 0s, Duration = Max	51.0%	53.6%	42.6%	59.2%	55.4%	43.6%	57.0%	51.8%	**57.2%**
Offset = 0s, Duration = MaxWithTrim	52.0%	53.8%	53.6%	61.0%	56.2%	53.6%	57.0%	55.3%	**58.1%**

Table 4. Tuning of the offset and duration.

4.4 Tuning of data preprocessing and model parameters

Given that the optimal feature extraction parameters had been gathered, the next step was to test if a different normalization could lead to improved classification accuracies. Another commonly used normalization method in machine learning is min-max, which sets the normalized scale for a feature from the instances with the largest and smallest values for that given feature. The differences in the outcome of the classification were however close to none. For some of the models, the results were lower, which may be a result from min-max giving more weight for outliers. The results are shown in Table 5 below.

Grid 3: Tuning of the normalization method									
Current Parameters	Song amount = 500, Sampling rate (Hz) = 15000, Offset (s) = 30, Song duration (s) = 150, Normalization = ?, n_features = 52, Model parameters = GridSearchCV with test parameters, Validation = StratifiedKFold (n_splits = 10)								
Model / Tuning	**Gaussian Naive Bayes**	**k-NN**	**Decision Tree**	**SVM's**	**Random Forests**	**AdaBoost**	**Neural Networks**	Average (Total)	Average (Top 3)
Normalization = Z-scores	54.4%	53.6%	46.0%	62.8%	60.0%	53.4%	58.0%	55.5%	**60.3%**
Normalization = Min-max	54.4%	49.2%	46.0%	61.6%	60.0%	53.4%	51.6%	53.7%	**58.7%**

Table 5. Tuning of the normalization method.

To account for the curse of dimensionality, the number of features was sequentially scaled down with the extremely randomized trees (or ExtraTrees) classifier, which is a commonly used embedded feature selection method. ExtraTrees, which was originally developed by Geurts et al. (2006), resembles the Random Forests algorithm. Unlike Random Forests however, ExtraTrees eschews bootstrapping and uses randomized cut points instead of cut points that maximize the gain. The ExtraTrees classifier provides a score that shows the relative importance of each feature during the construction of the decision trees. The goal was to test if any improvements could be achieved by omitting any redundant features. Using all the 52 features yielded the best results on average, but almost as accurate results could be achieved by using subsets of the 20 and 30 best features, as shown in Table 6 below.

Grid 4: Tuning of the number of features									
Current Parameters	Song amount = 500, Sampling rate (Hz) = 15000, Offset (s) = 30, Song duration (s) = 150, Normalization = Z-scores, n_features = ?, Model parameters = GridSearchCV with test parameters, Validation = StratifiedKFold (n_splits = 10)								
Model / Tuning	**Gaussian Naive Bayes**	**k-NN**	**Decision Tree**	**SVM's**	**Random Forests**	**AdaBoost**	**Neural Networks**	Average (Total)	Average (Top 3)
n_features = 52	54.4%	53.6%	46.0%	62.8%	60.0%	53.4%	58.0%	55.5%	**60.3%**
n_features = 40	53.2%	53.6%	45.0%	60.0%	52.4%	53.2%	58.0%	53.6%	**57.2%**
n_features = 30	52.2%	55.4%	52.2%	61.4%	57.8%	54.2%	58.0%	55.9%	**59.0%**
n_features = 20	53.4%	56.4%	47.2%	61.6%	57.0%	54.0%	58.6%	55.5%	**59.1%**
n_features = 10	49.6%	53.6%	44.2%	56.4%	52.8%	53.6%	52.4%	51.8%	**54.5%**

Table 6. Tuning of the number of features.

After the parameters related to data preprocessing were tuned, an attempt of reaching even higher classification accuracies was attempted by further tuning the parameters of the learning algorithms. So far, the test parameters have been tried out with GridSearchCV, which is an exhaustive search over specified model parameter values for an estimator. In other words, the model is trained and validated for ten times with every possible model parameter combination provided within the parameter grids. In the end, the optimal model parameters are gathered

for the final model to be used. By extending the parameter grid, it was hypothesized that some improvement could have been achieved. However, extending the parameter grid did not yield any higher classification accuracies, which means that the parameters used in the test grid turned out to capture the variations in the data in the best possible way. The SVM with a radial basis function kernel achieved the highest classification accuracy of 62.8%. The computationally more efficient ensemble method Random Forests scored a classification accuracy of 60.0%, followed by the Neural Network, which achieved an average classification accuracy of 58.0% over the 10 cross-validated folds.

The optimal model parameters are shown in Table 7. If a parameter is not specified, then it uses the default parameter provided by Python v3.6, Scikit-learn v0.19.2 or Keras v2.2.4.

Model	***Optimal Parameters***	***Score (%)***
Gaussian Naïve Bayes	-	54.4%
k-NN	*{n_neighbors: 8}*	53.6%
Decision Tree	*{'criterion': 'gini', 'max_depth': 11, 'min_samples_split': 30, 'splitter': 'random'}*	46.0%
SVM's	*{'C': 50, 'gamma': 0.001, 'kernel': 'rbf'}*	62.8%
Random Forests	*{'bootstrap': 'True', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_split': 3, 'n_estimators': 100}*	60.0%
AdaBoost	*{'algorithm': 'SAMME', 'learning_rate': 1, 'n_estimators': 50}*	53.4%
Neural Networks	Input dimension: 52, Hidden layer 1: 30 neurons, Hidden layer 2: 15 neurons, Epochs: 100, Batch size: 10, Optimizer: Adam, Loss function: Categorical Cross Entropy, Early Stopping: 5	58.0%

Table 7. The optimal parameters for the models.

4.5 Result Analysis

After the optimal models were attained, the next step was to derive even more insights into their results. Firstly, it is of interest to know which of the features were the most valuable predictors. The ExtraTrees embedding method for feature selection was used, which provides the relative feature importance scores. Figure 18 below shows that the five most valuable features were the variances of the highest MFCC's. Moreover, variance as an aggregation method tends to outperform means as an aggregation method.

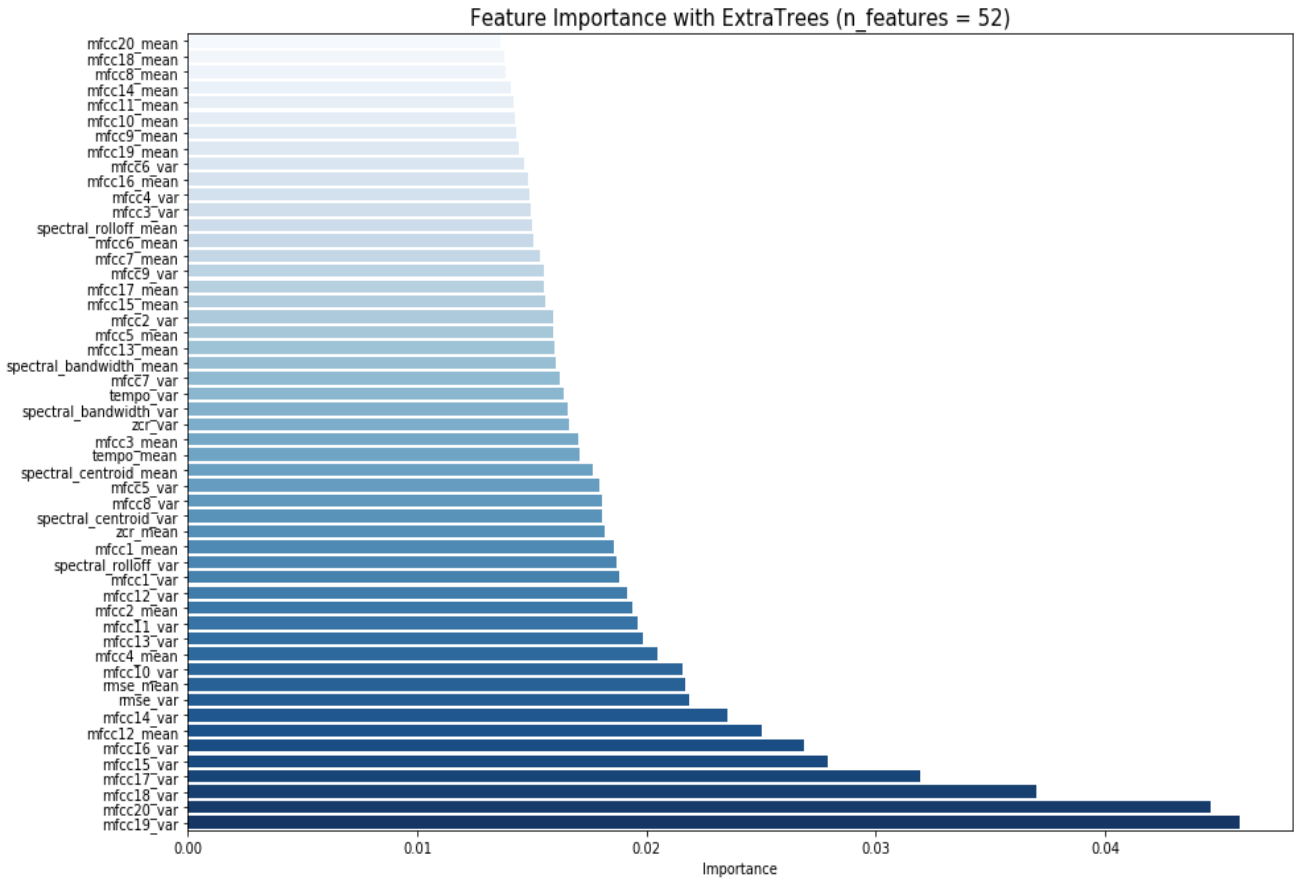

Figure 18. Feature importance for all the aggregated 52 features, created with the ExtraTrees embedding method.

Unsupervised learning methods were applied to achieve a visual understanding of the actual observations in the data set. More specifically, dimensionality reduction through PCA was used. The 52 features were first transformed to their corresponding eigenvectors. The two eigenvectors with the highest eigenvalues (i.e. principal component 1 and principal component 2) were then plotted as a scatterplot.

The results from the dimensionality reduction can be seen in Figure 19 below. The capital letters depict the class centroids, which were derived by taking the means of the principal components within each genre. The fuzziness of the subgenres is noticeable, since overlapping seems to happen to a large extent, even though muddy clusters are to some extent noticeable. The biggest overlapping seems to occur between death metal and thrash metal, while folk metal and heavy metal seem to have the largest spread. Moreover, some outliers are noticeable. A further check on the outliers were done by listening to the specific audio files, but no anomalies were found.

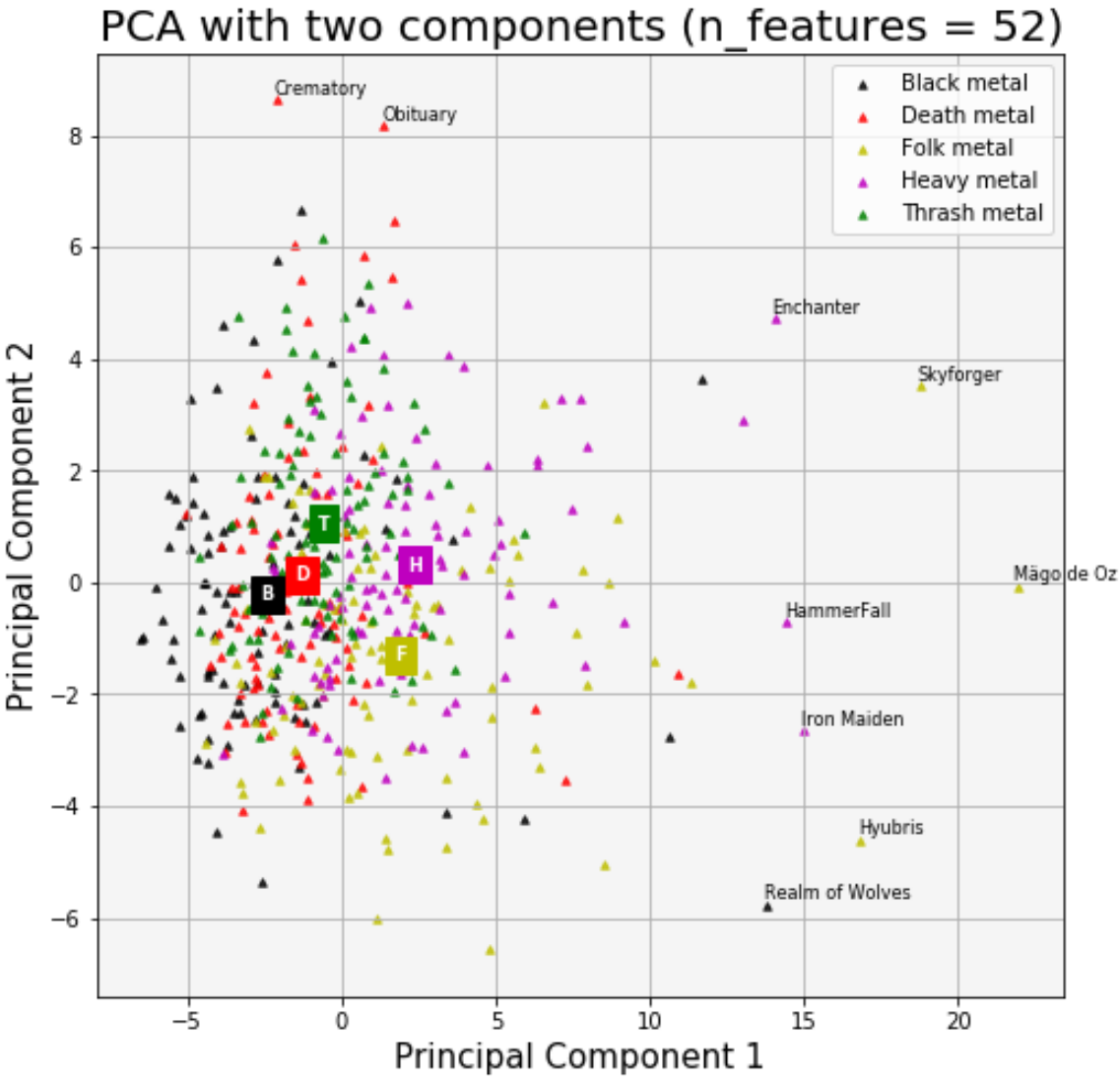

Figure 19. Scatterplot for PCA with two components where each song is depicted by a triangle. The centroids are depicted by the capital letters.

By examining the training history of the neural network in Figure 20, it can be observed that the model learns the pattern of the data for this fold set quickly. When evaluated on the training set, the network achieves a perfect classification accuracy of 100%. When evaluated on test set, the accuracy stagnates rather quickly around 10 epochs (i.e. when the training set has been looped over for 10 iterations), which indicates overfitting. Since accounting for the bias-variance tradeoff by lowering the number of neurons for a more simplified network did not yield positive effects, more high-quality data or more efficient features are likely required to acquire a greater true generalization rate.

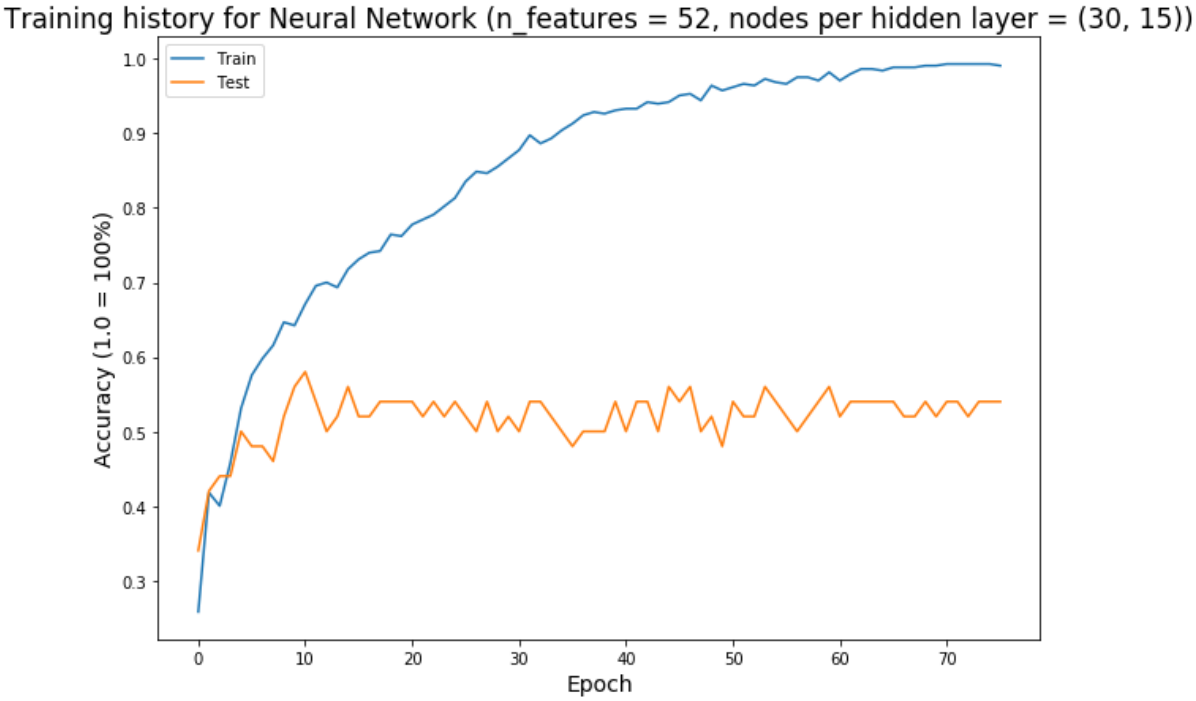

Figure 20. Training history of a single fold of the artificial neural network.

Next, it is of interest to know which genres were most often confused with each other across the models. Figures 21A-21G show the percentual confusion matrices for each of the models in the order of complexity. The x-axis depicts the actual observations, while the y-axis depicts the predicted observations.

Black metal seems to have been the easiest subgenre to classify, possibly because of its distinguishable and high timbral brightness. Black metal was most often confused with death metal. This could be a result of melodic death metal songs being also included under the subgenre of death metal, since the screamed vocals in melodic death metal can resemble those used in black metal.

Heavy metal seems to have been the second easiest subgenre to classify. It was most often mixed with the subgenres that it inspired, thrash metal and folk metal. Likewise, death metal was most often confused with thrash metal and vice versa, which seems reasonable, since death metal evolved from thrash metal.

Surprisingly, folk metal was not often confused with black metal which it has said to have evolved from. This was hypothesized beforehand, since the growls in folk metal are often alike the ones used in black metal. Instead, folk metal was most often confused with heavy metal, which it is also said to have developed from. Another reason for folk metal being prone to be classified as heavy metal could be that the spectral energy of the acoustic parts in folk metal is easier mixed with heavy metal, since the spectral energy of the three other subgenres is too high to be easily mixed with.

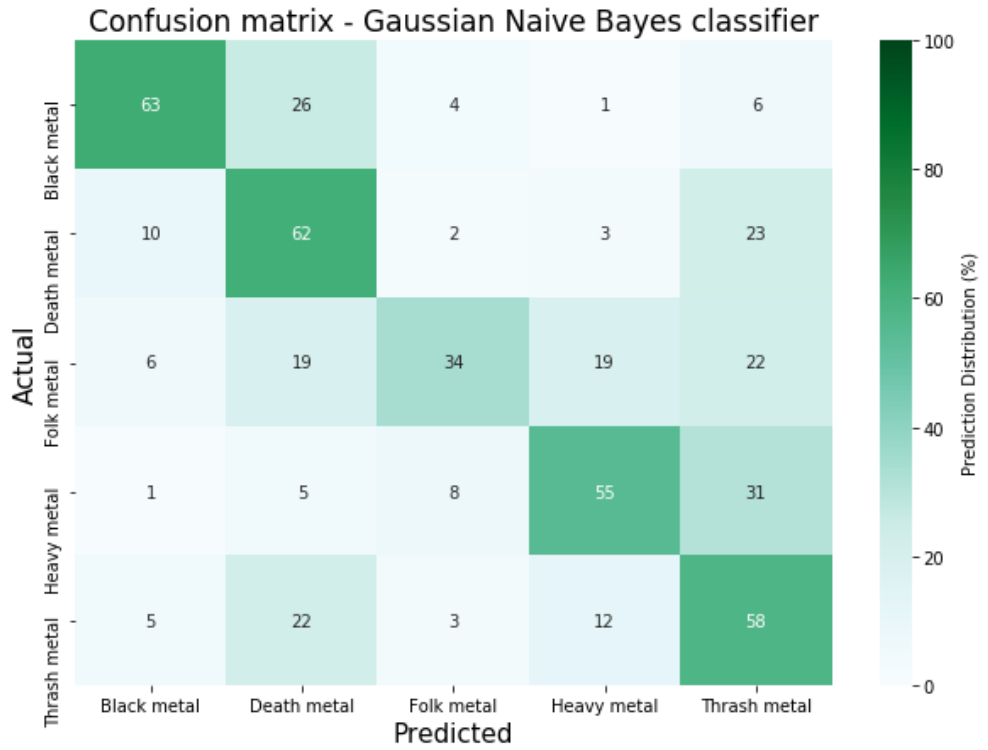

Figure 21A. Percentual confusion matrix for the Gaussian Naïve Bayes model.

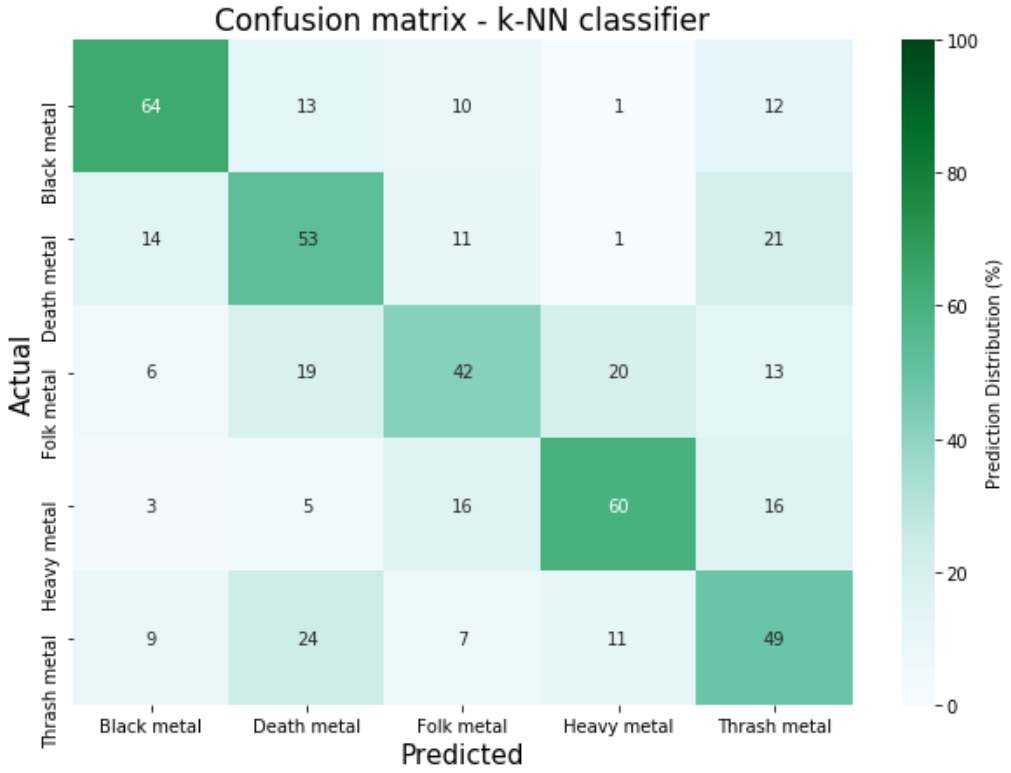

Figure 21B. Percentual confusion matrix for the k-NN model.

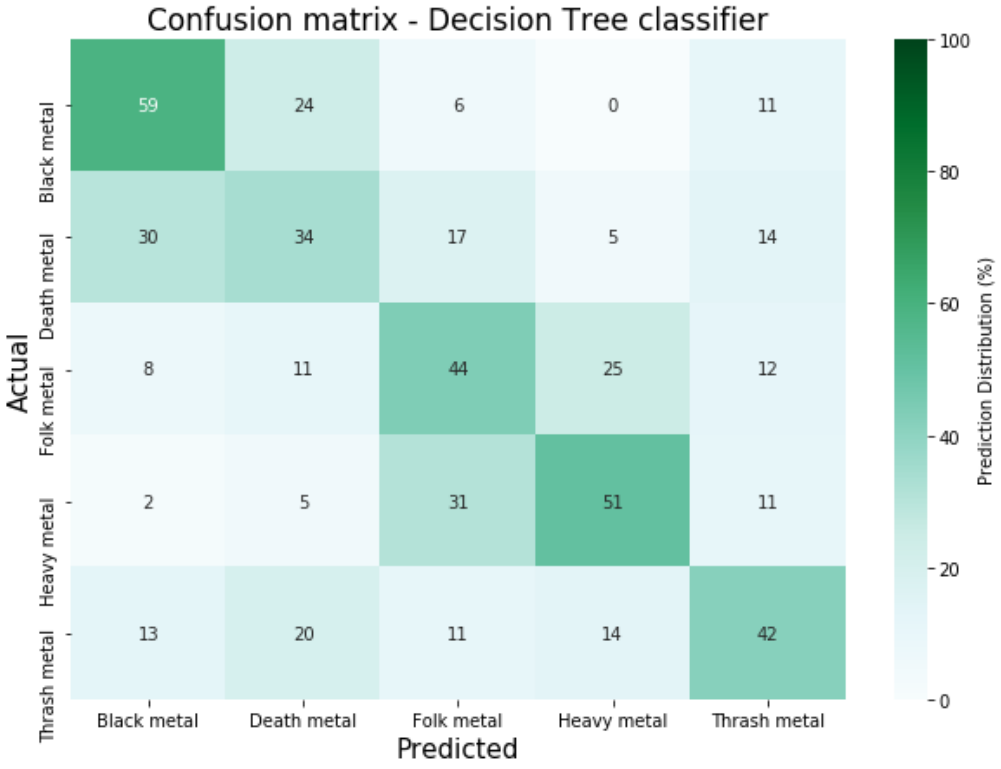

Figure 21C. Percentual confusion matrix for the Decision Tree model.

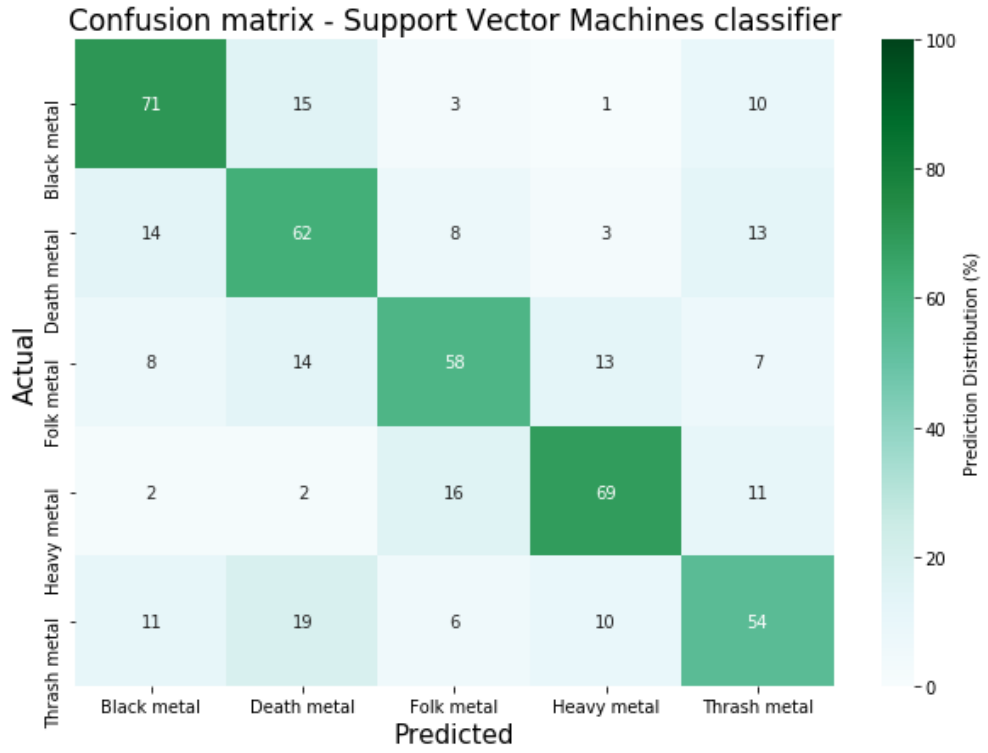

Figure 21D. Percentual confusion matrix for the SVM's model.

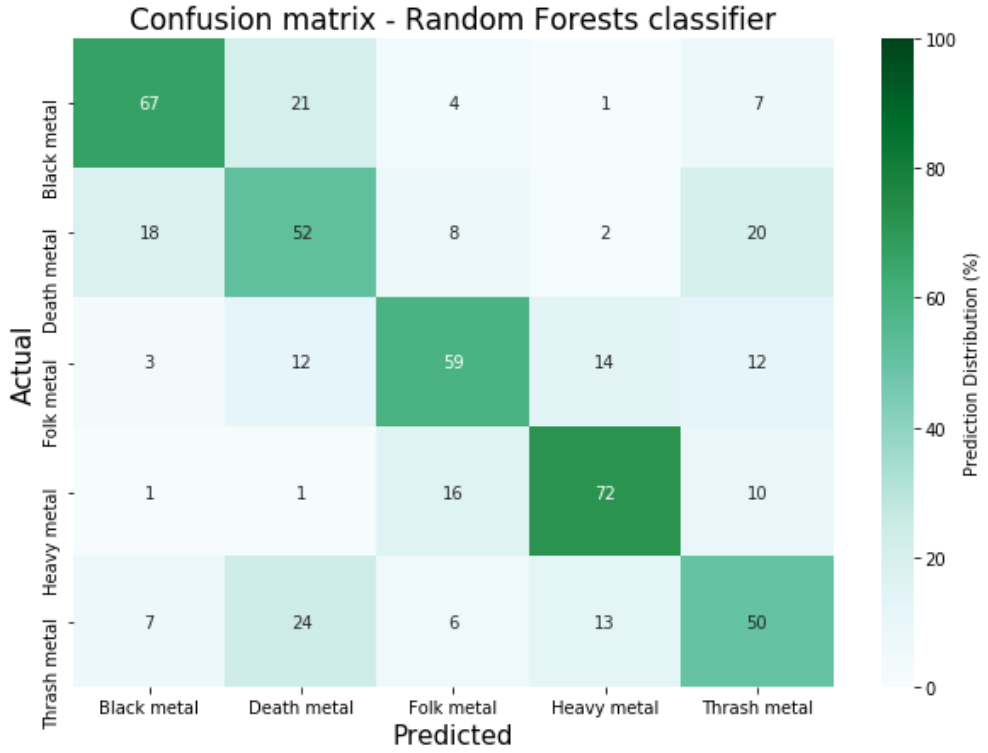

Figure 21E. Percentual confusion matrix for the k-NN model.

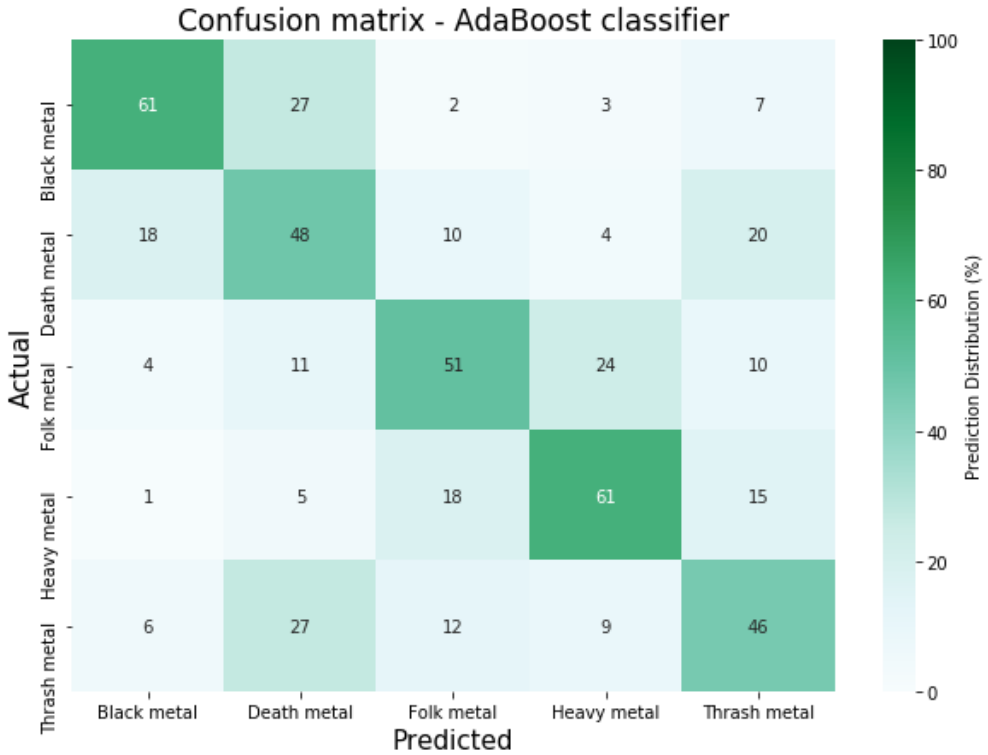

Figure 21F. Percentual confusion matrix for the AdaBoost model.

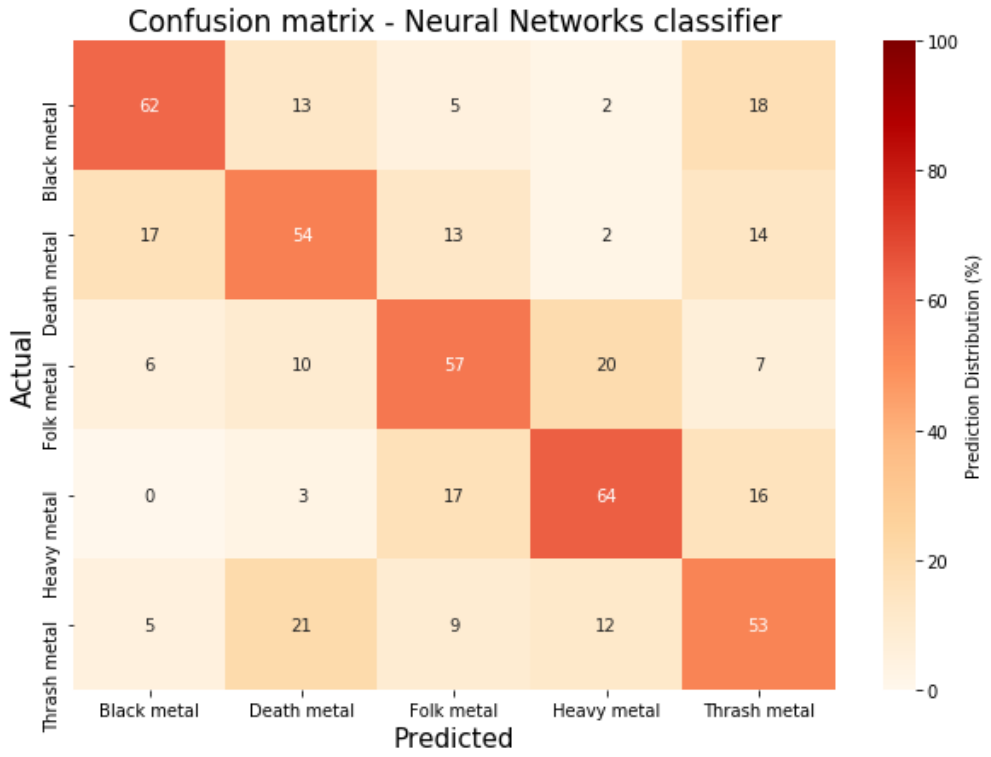

Figure 21G. Percentual confusion matrix for the Artificial Neural Network.

It is also of interest to know if some songs were more prone to be misclassified than others across all the models, or if the misclassification counts were relatively normally distributed. Since each of the 500 songs were included exactly once in the test set across the seven models, a prediction was made seven times on each of the 500 songs. Thus, the total amount of times an input vector was misclassified can be counted.

The song-wise misclassification counts across the seven models are shown as a stacked bar chart in Figure 22A, where each stacked bar corresponds to a genre. Seven misclassifications would indicate that a song failed across all the seven models during its appearance in the test set. A stacked bar with a height of 60 would instead indicate that 60 of the songs within a genre were misclassified at least once, which would in turn mean that 40 of the songs were correctly classified for all the models. Figure 22A confirms that black metal was the easiest genre to classify, since 30 of the 100 songs were correctly classified by all seven models. In general, about 5 to 10 percent of the songs within a genre were misclassified across all seven models.

Similarly, Figure 22B shows the song-wise misclassification for the three best models. By focusing on the three best performing models, it can be observed that approximately 15 to 20

percent of the songs within each of the genres were misclassified three times. Interestingly, approximately 60 percent of the heavy metal songs and 45 percent of black metal songs were correctly classified across all three best performing models, which would indicate that heavy metal was the easiest genre to classify among the three best performing models.

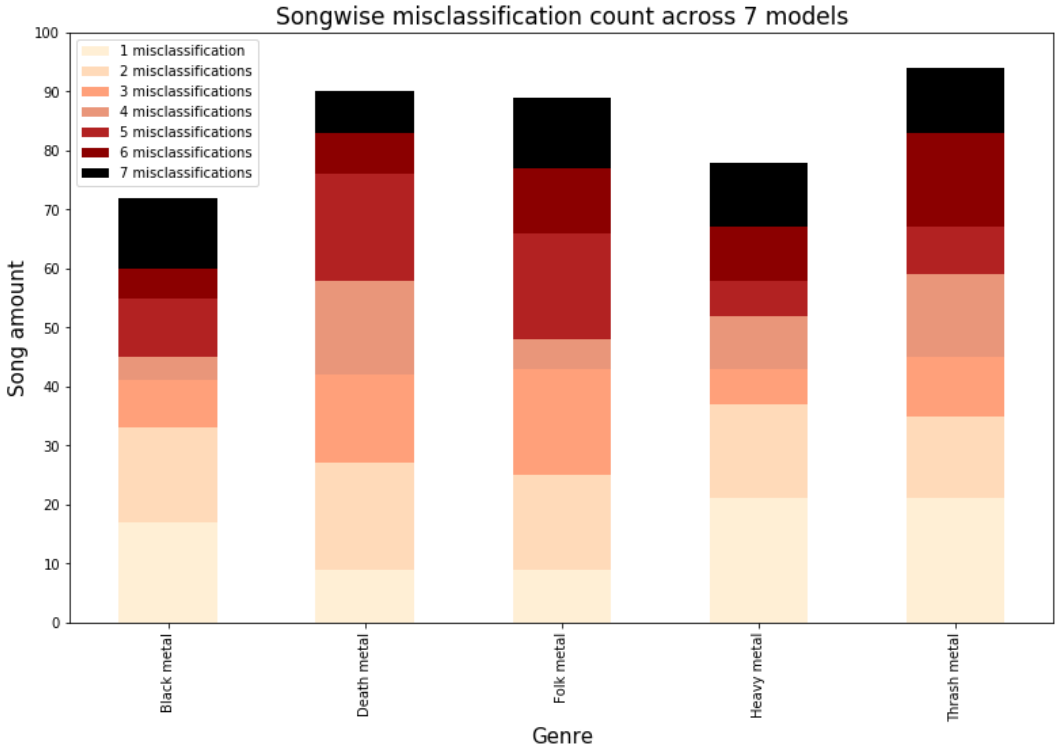

Figure 22A. Song-wise misclassification across all seven models.

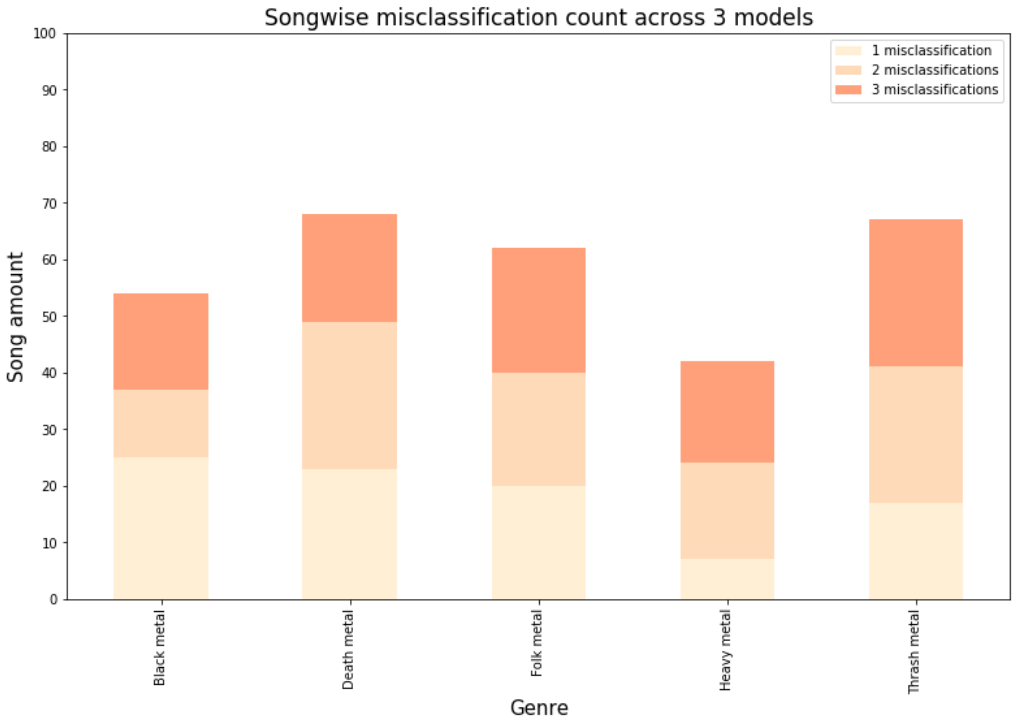

Figure 22B. Song-wise misclassification across the three best models (SVM's, Random Forests and Artificial Neural Networks).

Finally, the classification reports for each of the seven models can be shown in Tables 8A-8G. The tables show genre-wise and averaged precision, recall and F_1-scores. The support columns stand for the number of songs included in the analysis.

Gaussian Naïve Bayes					
	Precision	Recall	F1-Score	Support	Accuracy
Black Metal	0.74	0.63	0.68	100	
Death Metal	0.46	0.62	0.53	100	
Folk Metal	0.67	0.34	0.45	100	
Heavy Metal	0.61	0.55	0.58	100	
Thrash Metal	0.41	0.58	0.48	100	
Avg / Total	**0.58**	**0.54**	**0.54**	**500**	**54.4%**

Table 8A. Classification report for the Gaussian Naïve Bayes classifier.

k-NN					
	Precision	Recall	F1-Score	Support	Accuracy
Black Metal	0.67	0.64	0.65	100	
Death Metal	0.46	0.53	0.50	100	
Folk Metal	0.49	0.42	0.45	100	
Heavy Metal	0.65	0.60	0.62	100	
Thrash Metal	0.44	0.49	0.46	100	
Avg / Total	**0.54**	**0.54**	**0.54**	**500**	**53.6%**

Table 8B. Classification report for the k-NN classifier.

Decision Tree					
	Precision	Recall	F1-Score	Support	Accuracy
Black Metal	0.53	0.59	0.56	100	
Death Metal	0.36	0.34	0.35	100	
Folk Metal	0.40	0.44	0.42	100	
Heavy Metal	0.54	0.51	0.52	100	
Thrash Metal	0.47	0.42	0.44	100	
Avg / Total	**0.46**	**0.46**	**0.46**	**500**	**46.0%**

Table 8C. Classification report for the Decision Tree classifier

Support Vector Machines					
	Precision	Recall	F1-Score	Support	Accuracy
Black Metal	0.67	0.71	0.69	100	
Death Metal	0.55	0.62	0.58	100	
Folk Metal	0.64	0.58	0.61	100	
Heavy Metal	0.72	0.69	0.70	100	
Thrash Metal	0.57	0.54	0.55	100	
Avg / Total	**0,63**	**0.63**	**0.63**	**500**	**62.8%**

Table 8D. Classification report for the SVM's classifier.

Random Forests					
	Precision	Recall	F1-Score	Support	Accuracy
Black Metal	0.70	0.67	0.68	100	
Death Metal	0.47	0.52	0.50	100	
Folk Metal	0.63	0.59	0.61	100	
Heavy Metal	0.71	0.72	0.71	100	
Thrash Metal	0.51	0.50	0.50	100	
Avg / Total	**0.60**	**0.60**	**0.60**	**500**	**60.0%**

Table 8E. Classification report for the Random Forests classifier.

AdaBoost					
	Precision	Recall	F1-Score	Support	Accuracy
Black Metal	0.68	0.61	0.64	100	
Death Metal	0.41	0.48	0.44	100	
Folk Metal	0.55	0.51	0.53	100	
Heavy Metal	0.60	0.61	0.61	100	
Thrash Metal	0.47	0.47	0.46	100	
Avg / Total	**0.54**	**0.53**	**0.54**	**500**	**53.4%**

Table 8F. Classification report for the AdaBoost classifier.

Artificial Neural Network					
	Precision	Recall	F1-Score	Support	Accuracy
Black Metal	0.69	0.62	0.65	100	
Death Metal	0.53	0.54	0.54	100	
Folk Metal	0.56	0.57	0.57	100	
Heavy Metal	0.64	0.64	0.64	100	
Thrash Metal	0.49	0.53	0.51	100	
Avg / Total	**0.58**	**0.58**	**0.58**	**500**	**58.0%**

Table 8G. Classification report for the ANN classifier.

5. Discussion

It is reasonable that relatively new research fields can, atleast to some extent, be negatively affected by complicated factors, that can skew the research results. This is particularly true when two multidimensional fields, such as MIR and machine learning interconnect. For scientific purposes, re-evaluation of previous research may be needed. This study managed to contribute to the re-evaluation of the current AMGC-related research within the field of MIR, which has been plagued with various integrity issues. This was achieved by creating a custom-made data set, which focused on the restricted domain of heavy metal music, while also accounting for all integrity issues. The relatively lower performance metric scores from this study indicates that state-of-the-art results are often quite overoptimistic.

For successive real-life value creation with AMGC-systems, more high-quality data are needed to realistically solve the problem of insufficiently labeled musical data within large databases, where genres may dynamically change over time. The complexity of polyphonic signals makes the task of classification challenging, since insufficient feature engineering methods lead to noise being included in the form of hidden features. The solution could lie in factors, such as more sophisticated feature engineering methods, aggregation methods, signal transformation methods or machine learning models. Even though the nature of the task is challenging, a successive AMGC-system should still be strived for. Such a system could provide great business value for streaming services by enabling sophisticated methods for personalized marketing.

The research questions for this study were:

1. Can machine learning be used to distinguish between subgenres of heavy metal music?

2. What is the most effective learning algorithm for distinguishing between subgenres of heavy metal music?"

3. What insights can be derived from the outcome of the process?

To answer the first research question whether machine learning can be used to distinguished between subgenres of heavy metal is subjective. Given that guessing by chance should have resulted in classification accuracies of approximately 20% with five subgenres, the best result of 62.8% would indicate that the AMGC-system works quite admirably.

The second research question concerning which was the most effective learning algorithm, the answer is SVM's with a radial basis function kernel, which scored a classification accuracy of 62.8% on average over the ten folds. This was closely followed by the computationally more efficient ensemble method Random Forests (60.0%) and the Artificial Neural Network (58.0%). The more interpretable models Gaussian Naïve Bayes and k-NN also performed quite admirably by providing average classification accuracies of 54.4% and 53.6%, respectively. The ensemble method AdaBoost reached a classification accuracy of 53.4 % on average over the test folds, while the averaged classification accuracy for the decision tree was only 46.0%.

Concerning the final research question, there are several insights to be made from the study. Firstly, the sampling rate seems to play a rather large role in the process of AMGC, maybe more than what has been discussed before. This was shown during the tuning of the feature extraction parameters. Counterintuitively, a too high sampling rate may affect the learning in a negative way, and the generally accepted default value of 22050 Hz may not be the optimal one. Secondly, the song data can be generalized from a rather short excerpt, which may be of interest when building AMGC-systems from large databases. This was shown during the tuning of the feature extraction parameters, when comparing the use of full audio signals with the shorter song excerpts of 150 seconds. Thirdly, a manual offset may be of importance for future AMGC-systems, since it was shown to raise the classification accuracies during the tuning of the feature extraction parameters. This may be due to its ability to exclude potential non-relevant spectral activity of various sorts from the beginning of songs. Fourthly, variance as a feature aggregation method should be preferred over means. This was shown from the relative feature importance scores by using the ExtraTrees embedded feature selection method. Fifthly, the upper MFCC's are powerful for distinguishing between subgenres of heavy metal, which was also shown from the relative feature importance scores.

Probably the most important insight that can be derived from the results is, however, that more high-quality data in the form of a larger number of useful features should be included in future models for improved results. Since the lyrics across the subgenres often have their own niches (i.e. black metal tends to have a focus on topics such as religion and nature, thrash metal has a tendency towards societal critique and war, folk metal has a tendency of being focused on mythology and history, while the lyrics in death metal can be graphically exceedingly violent), one possibility could be to make use of natural language processing

methods. For instance, a bag-of-words or word embedding model could be created to derive content-based lyrical features. In addition to this, a method of improvement could be to include some form of tag-related data from other sources.

Finally, improvements might also be achieved by developing even more complex deep learning models and letting them work together. Several models could, for example, be utilized together by letting them cast votes for target labels. As an example, feeding sequential non-aggregated data as input to a long short-term memory comprised RNN could be attempted, which would interestingly also utilize reinforcement learning as a learning type. In parallel to this, spectrograms could be directly fed to a CNN.

6. Svensk sammanfattning: Klassificering av heavy metal-subgenrer med maskininlärning

6.1 Introduktion

I det nutida samhället genereras allt större mängder data inom olika industrier. Ur en värdeskapande synvinkel har dessa så kallade Big Data visat sig erbjuda stora möjligheter. Samtidigt råder det utmaningar i hur dessa data ska bearbetas och tolkas på ett meningsfullt sätt. Som en respons till dessa svårigheter har olika tekniker inom artificiell intelligens (AI) vuxit fram. En av dessa tekniker är maskininlärning, vilket är en samling metoder som automatiskt upptäcker mönster i stora datamängder. Utöver detta har ett delområde, djupinlärning, utvecklats inom forskningsfältet av maskininlärning. Djupinlärning är en form av maskininlärning som använder sig av algoritmer som inspirerats av hjärnans biologiska struktur och dess funktioner.

Fram till nutid har dessa AI-tekniker främst använts inom tillverknings-, försäljnings-, hälsovårds- och finansindustrin. Under de senaste åren har dessa tekniker dock även vuxit fram inom musikindustrin genom innovativa företag så som Shazam och SoundHound. Framväxten begränsar sig inte endast till musikaliska innovationer, utan musikindustrins struktur genomgår också för tillfället en transformation i en mer sofistikerad riktning. Denna utveckling möjliggör en anknytning mellan försäljning och marknadsföring, eftersom dessa så kallade AI-drivna strömningsdata från företag så som Spotify och YouTube kan tillåta en djup förståelse av konsumtionsmönster. Eftersom konsumtionsmönstren dock varierar utifrån olika faktorer så som nivå av engagemang och favoritgenre, bör unika marknadsföringsstrategier riktas gentemot kundsegmenten för maximal effektivitet.

En utmaning med denna genrebaserade segmenteringsstrategi ligger i nutidens väldigt stora databaser, vars upprätthållande har bevisats kräva omfattande mängder arbete och tid. Som en respons på denna utmaning har forskningsfältet av musikinformationssökning (eng. Music Information Retrieval - MIR) etablerats under 2000-talet. Huvudvikten inom MIR ligger i utvinning av information genom digital signalbehandling. Information kan även utvinnas genom sådana musikdata som inte direkt kan förknippas med ljudsignalen (t.ex låttexter, användarrecensioner, låtnoter och bibliografisk information). Ett av de mest populära forskningsområdena inom MIR är automatisk klassificering av musikgenrer (eng. Automatic music genre classification – AMGC). De två huvudsakliga uppgifterna inom AMGC är

effektiv utvinning och selektion av variabler, samt applicering av maskininlärningsalgoritmer för effektiv klassificering.

6.2 Problematisering

Medan MIR som forskningsfält är relativt nytt, har speciellt den AMGC-relaterade forskningen inom fältet hunnit drabbas av stora integritetsproblem. Orsaken till detta är fältets mest använda dataset, GTZAN. Sedan datasetet introducerades år 2002 har det fram till år 2013 inkluderats i åtminstone 100 publicerade artiklar, vilket utgör ungefär 40 % av den totala AMGC-relaterade forskningen. Datasetet består av 1000 stycken halvminuter långa låtklipp bland 10 jämnfördelade genrer, samt av låtdata i form av variabler på lågnivå, som utvunnits från dessa låtklipp.

De bevisade integritetsproblemen i datasetet förekommer i form av exakta repetitioner, inspelningsrepetitioner, artistrepetitioner och versionrepetitioner. Exakta repetitioner, som hittats vid 50 fall, innebär att nästan identiska låtklipp använts mer än en gång. Inspelningsrepetitioner, som hittats vid 21 fall, innebär att fler än ett låtklipp använts från en låt. Artistrepetitioner, som förekommer i nästan alla klipp, innebär att en artists låtklipp använts fler än en gång. Exempelvis består genren reggae till mer än en tredjedel av Bob Marleys musik. Versionrepetitioner, som hittats vid 13 fall, innebär att ett låtklipp använts från mer än en version av samma låt. En coverversion eller en remix kan till exempel ha använts tillsammans med den ursprungliga studioversionen.

Utöver dessa integritetsproblem har datasetet kritiserats för dess brist av metadata, samt för faktumet att ungefär 10 % av låtarna är kategoriserade enligt fel genre. Cirka 2 % av låtarna har dessutom visat sig innehålla kvalitetsproblem i form av t.ex distorsion. Dessutom innehåller datasetet bevisligen inga subgenrer, eftersom de uppgivna genrerna är blues, klassisk, country, disco, hip hop, jazz, metal, pop, reggae och rock. Alla dessa ovannämnda faktorer bidrar till överoptimistiska resultat i klassificeringsprocessen. Klassificeringsnoggrannheterna ligger vanligtvis mellan 61 och 82 % för detta dataset, men till och med noggrannheter på 90 % har rapporterats.

6.3 Syfte

Givet att den nuvarande forskningen drabbats av integritetsproblem till en så pass hög grad är målet med denna pro gradu-avhandling, *”Classification of Heavy Metal Subgenres with Machine Learning”*, att återevaluera användandet av maskininlärningsmetoder för AMGC.

För det första kommer alla de presenterade integritetsproblemen, som drabbat åtminstone 40 % av forskningen fram till 2013, att beaktas. För det andra kommer avhandlingen att begränsa sig till subgenrer av heavy metal-musik, eftersom användning av en begränsad domän i tidigare forskning rekommenderats som metod för förbättrade klassificeringsresultat. Utöver detta är de suddiga gränserna mellan de olika subgenrerna av heavy metal ett effektivt sätt att på ett tillförlitligt vis utföra återevalueringen. Resultaten av denna avhandling kan vara av intresse för strömningsföretag, skivbolag och marknadsföringsföretag inom musikindustrin.

De tre forskningsfrågorna är:

1. Kan maskininlärning användas till urskiljning av subgenrer av heavy metal-musik?

2. Vilken är den mest effektiva inlärningsalgoritmen för urskiljandet av subgenrerna?

3. Vilka insikter kan härledas från processen?

6.4 Metod

För att besvara den första forskningsfrågan kommer variabler av lågnivå att utvinnas från totalt 500 insamlade ljudfiler. Ljudfilerna består av fem jämnfördelade genrer, vilket innebär att varje genre representeras av 100 ljudfiler från en personlig musiksamling. Subgenrerna valdes med omsorg att representera hela spektrumet av metal-musik. De valda subgenrerna är heavy metal, thrash metal, death metal, black metal och folk metal. Samtliga låtar inom en subgenre har en minimilängd på tre minuter, men ingen maximilängd.

För att beakta alla integritetsproblem (dvs. exakta repetitioner, inspelningsrepetitioner, artistrepetitioner och versionrepetitioner), bestämdes att det totala antalet unika artister är lika med 500. För att redogöra för potentiella kvalitetsproblem och andra anomalier, kontrolleras alla ljudfiler manuellt. Slutligen, för att redogöra för problemet med felklassificerade genrer kommer samtliga ljudfiler att manuellt klassificeras av mig, som har uppskattningsvis 14 års erfarenhet av olika subgenrer av heavy metal-musik. Med hjälp av de utvunna lågnivåvariablerna och olika inlärningsalgoritmer, konstrueras och evalueras olika maskininlärningsmodeller för ett AMGC-system. Eftersom klasserna är balanserade, används klassificeringsnoggrannhet och förväxlingsmatriser som de primära metoderna vid evalueringsprocessen. Utöver detta kommer även precision, recall och F_1 anges, så att resultaten ska kunna jämföras med framtida forskning. Av klassificeringsresultaten kan en

slutsats dras gällande om klassificeringen av heavy metal-subgenrer med maskininlärningsmetoder var lyckad eller ej.

För att besvara den andra forskningsfrågan kommer de applicerade maskininlärningsmodellerna att jämföras med varandra utifrån klassificeringsnoggrannheten som uppnåddes med en inlärningsalgoritm. Inlärningsalgoritmerna som valdes till denna avhandling är sex av de möjligtvis mest kända algoritmerna för ett klassificeringsproblem: Naïve Bayes, K-Nearest Neighbors, beslutsträd (eng. Decision Trees), stödvektormaskiner (eng. Support Vector Machines – SVM's), Random Forests och AdaBoost. Utöver detta kommer ett artificiellt neuronnät (eng. Artificial Neural Networks – ANN's) att inkluderas, vilket faller under delområdet djupinlärning. Detta resulterar i att totalt sju olika inlärningsalgoritmer kommer i slutändan att rangordnas för att uppnå en slutsats om vilken algoritm som fungerade bäst för det givna problemet.

För att besvara den tredje forskningsfrågan kommer olika insikter att försöka härledas från processen. Till exempel kan det vara av intresse att veta vilka parametrar som fungerade bäst under variabelutvinningen, vilka parametrar som fungerade bäst för maskininlärningsmodellerna, eller vilka genrer som oftast blandades ihop med varandra. Angående de tekniska detaljerna kommer programmeringsspråket Python, ljudanalysmodulen Librosa, maskininlärningsmodulen Scikit-learn, samt djupinlärningsmodulen Keras med Tensorflow som back-end att vara de primära verktygen för studien.

6.5 Variabelutvinning och MIR

Från en teknisk synvinkel kan musik representeras på tre sätt: partitur (dvs. notblad), symbolisk musik (t.ex MIDI-filer) och ljudfiler (t.ex .WAV, .MP3). En utmaning i att utvinna meningsfulla data från ljudfiler är att parametrarna i dessa inte är explicit givna, vilket de i sin tur är i partitur och symbolisk musik. Istället har alla komponenter under inspelningen resulterat i en kombinerad ljudvåg, vilket innebär att utvinning av meningsfull information är utmanande. Denna utmaning i sin tur leder till utmaningar i själva klassificeringsprocessen för en maskin.

En ljudfil kan representeras i tidsdomänen som vågform, men vågformer är inte gynnsamma ur perspektivet av variabelutvinning. Den enklaste typen av vågform kallas sinusvåg. Dessa vågor är statiska, kontinuerliga och oändliga. Sinusvågor anges i Hertz (Hz), där Hz definieras som inversen av antal utförda cykler som vågformen gör under en sekund. Inom musik är

vågformerna dock mycket mer komplexa och de indelas oftast in i monofoniska och polyfoniska signaler. I en monofonisk signal spelas en not åt gången, medan i polyfoniska signaler spelas mer än en not åt gången, vilket innebär att musik nästan uteslutet är polyfonisk.

Ljudvågor är inte användarvänliga för variabelutvinning, utan istället transformeras de från tidsdomänen till frekvensdomänen eller tidsfrekvensdomänen innan variabelutvinningen. Det möjligtvis mest populära sättet att utföra denna transformation är den matematiska formeln Short-Time Fourier Transform (STFT), som appliceras över korta överlappande ljudvågssnuttar eller ljudvågsramar. Outputen av STFT för en ändlig ljudvåg är en sekvens av vektorer, där varje vektor beskriver den spektrala energimängden av en ram. En enskild ram kan visualiseras i frekvensdomänen som ett spektrum, där x-axeln betecknar frekvensnivån och y-axeln betecknar energinivån.

För att möjliggöra visualiseringen av alla ramar på en gång används oftast ett spektrogram. I ett spektrogram betecknar x-axeln tiden, medan y-axeln betecknar frekvensen. Energinivån i sin tur betecknas på en färgskala. Inom musikanalys används spektrogrammen oftast på mel-skalan, vilken är en sorts logaritmisk skala som härmar den mänskliga hörseln. Denna specifika typ av spektrogram kallas mel-frekvens-spektrogram eller melspektrogram.

Efter att en ljudsignal omvandlats från tidsdomänen till den analysvänliga frekvens- eller tidsfrekvensdomänen kan meningsfulla variabler på lågnivå utvinnas. Dessa variabler är nödvändigtvis inte meningsfulla ur ett mänskligt eller musikologiskt perspektiv, men kan istället tolkas av maskiner. Exempel på lågnivåvariabler är kepstrala-koefficienter i Mel-frekvens (MFCC), spektral centroid, spektral bandbredd, spektral roll-off, kvadratiska medelvärdet, nollgenomgången och dynamiskt tempo. Eftersom dessa variabler anges separat för varje ram krävs aggregationsmetoder. Populära aggregationsmetoder är medelvärde och varians.

I kontrast till maskiners uppfattning av musik är den mänskliga uppfattningen av musik inte matematisk, utan subjektiv. Denna subjektivitet är av stor vikt, eftersom musikgenrer är helt och hållet människodefinierade. Utöver kulturella och emotionella aspekter skiljs olika genrer åt utifrån musikaliska faktorer så som instrumentation, rytmstruktur och klangfärg. Musikgenrer har dessutom bevisligen visat sig förändras över tid, vilket betyder att gränserna inom olika genrer kan vara väldigt suddiga. Komplexiteten av polyfoniska musiksignaler och

bristen på en universell genretaxonomi är seriösa utmaningar för automatisk klassificering av musikgenrer, speciellt då det handlar om suddiga subgenrer.

Till denna pro gradu-avhandling valdes subgenrerna heavy metal, thrash metal, death metal, black metal och folk metal. Heavy metal är en musikgenre som utvecklats kring slutet av 1960-talet. Heavy metal kan klassificeras som en subgenre av rockmusik. De fundamentala ljudliga dimensionerna i heavy metal är styrka, vilket uttrycks som högljuddhet via traditionella rockinstrument och en emotionellt intensiv sång. Heavy metal ledde till utvecklingen av thrash metal kring början på 1980-talet, där den främsta skillnaden är ett ökat tempo och en överlag mer aggressiv ljudvärld. Thrash metal ledde kring slutet av 1980-talet till utvecklingen av death metal, som är lättast urskiljbar genom den lågt rutna sångstilen, de lågstämda instrumenten och de tekniskt utmanande låtstrukturerna. Kring början på 1990-talet utvecklades black metal, som inspirerats till största del av både thrash metal och black metal. Den främsta karaktäristiken är en kakofonisk och diskant ljudvärld med gällskrikt sång. Folk metal har i sin tur spekulerats ha utvecklats under början av 1990-talet från både heavy metal och black metal. Den kännetecknas främst av inklusionen av traditionella folkinstrument i kontext av heavy metal.

6.6 Klassificering och maskininlärning

Inlärningssätten inom maskininlärning indelas oftast in i tre huvudtyper: övervakad inlärning, oövervakad inlärning och förstärkningsinlärning. I övervakad inlärning används förklarande variabler för att kartlägga sambandet gentemot en beroende variabel. Exempelvis kan en vektor, som består av aggregerade variabler på lågnivå och den korrekta subgenren för en viss låt användas som ett input, där målet är att lära sambandet om vilken subgenre låten hör till. Via generalisering borde maskinen sedan kunna applicera sin kunskap på att klassificera helt obekanta låtar inom ramar av de givna subgenrerna.

I kontrast till övervakad inlärning existerar inga rätta svar i oövervakad inlärning. Syftet med oövervakad inlärning är istället endast att finna intressanta mönster inom inputdata. Exempelvis kan underliggande grupper hittas inom ett stort antal komplexa observationer, vilket benämns som klustring. Dessutom kan oövervakad inlärning användas för densitetsestimering eller för att förminska dimensionerna av högdimensionsdata till ett två- eller tredimensionellt plan, så att visualiseringsmetoder ska kunna appliceras.

Förstärkningsinlärning används vid ramverk för robotik genom att låta en så kallad agent motta belöningar eller straff utifrån handlingen som agenten utför. Genom denna sk. ”försök och misstag”-metod kan agenten lära sig vilka handlingar är optimala enligt de givna kriterierna.

Den korrekta inlärningsmetoden för denna studie är övervakad inlärning, eftersom målet är att lära maskinen att känna igen subgenrer utifrån de aggregerade lågnivåvariablerna. Modelluppbyggnadsprocessen består av följande faser: variabelutvinning, utforskning och förberedning av data, modellselektion och finjustering av parametrar, modellvalidering och applicering av samplingsmetoder, beaktning av bias-varians tradeoffen, selektion av de optimala variablerna, samt resultrapportering med korrekta mått.

För det första har variabelutvinningen förklarats i kapitel 6.5. Denna fas är kritisk, eftersom utvinning av sådana oberoende variabler som starkt korrelerar med den förklarande variabeln ökar sannolikheten för succesiv inlärning. Denna fas är utmanande och kräver domänkunskap, intuition och matematiska transformationer. För det andra, efter att variablerna har utvunnits är följande steget att förstå sig på dess innehåll, så att de ska kunna vidarebearbetas och således förberedas för maskininlärningsmodeller. Exempelvis kan visuella metoder användas för att uppnå en förståelse av innehållet. Variablerna kan också behöva normaliseras, så att de får lika mycket vikt i maskininlärningsmodellerna. Felaktiga eller bristfälliga observationer kan behöva behandlas genom exempelvis exkludering.

Den tredje fasen är modellselektion och finjustering av parametrar. Detta är en iterativ fas, där modellen provas med olika parametrar för att kunna uppnå en förståelse av vilka parametrar som ledde till den mest successiva inlärningsnivån för den givna uppgiften. Den fjärde fasen är modellvalidering och applicering av samplingsmetoder. Syftet med denna fas är att på ett pålitligt sätt kunna bekräfta att modellen kan applicera sin kunskap på sådana data som modellen ännu inte sett, samt att successivitetsnivån inte beror på slumpen. Den mest populära samplingsmetoden inom maskininlärning är K-Fold Cross-Validation, där datasetet delas in i k antal delar. Ett vanligt värde för k är 10. Nio av delarna fungerar som träningsdata, medan den tionde delen fungerar som testdata. Maskinen tränas på träningsdatan och evalueras på testdatan. Processen utförs separat på nytt tills varje del har fungerat som testdata. Detta kommer att leda till ett pålitligt medelvärde för hur bra maskinen slutligen lyckades med uppgiften.

Den femte fasen består av beaktning av det sk. bias-varians-tradeoffet. Denna process består av att justera träningen av modellen till en sådan grad att den inte övertränas. Överträning leder till perfekta resultat vid evaluering på träningsdatan, men till svaga resultat då modellen försöker generalisera sin kunskap på testdata. Den sjätte fasen består av variabelselektion. Syftet med variabelselektion är att inklusion av ett stort antal icke-informativa variabler kan leda till försvagade resultat eller även misslyckad inlärning. Metoder som används i denna fas kan vara principalkomponentsanalys eller den inbäddade Extra Trees-metoden.

Slutligen, i den sjunde fasen bör korrekta mått användas för att evaluera successivitetsnivån av inlärningen. Klassificeringsnoggrannhet, dvs. det totala antalet korrekt klassificerade observationer dividerat med det totala antalet observationer i datasetet är ett av de vanligaste och mest intuitiva måtten. Denna metod har dock sina brister, eftersom inbalanser i klasserna kan leda till icke-tillförlitliga siffror, och resultaten bland olika studier är sällan jämförbara med detta mått. Ett annat populärt mått är förväxlingsmatrisen, som visar i vilken mån olika klasser blandades med varandra under klassificeringsprocessen. Andra kända mått är precision, recall och F_1, vilka beaktar potentiella inbalanser i klasserna.

Nyligen har de huvudsakliga faserna i modelluppbyggnadsprocessen förklarats. Slutligen bör en inlärningalgoritm väljas. Denna studie använder sig av sju olika inlärningsalgoritmer, vilka är Naïve Bayes, K-Nearest Neighbors, beslutsträd, stödvektormaskiner, Random Forests, AdaBoost och artificiella neuronnät, varav den sist nämnda kan kategoriseras som djupinlärning.

6.7 Empirisk studie

Det första steget med denna empiriska studie är anskaffning av de optimala parametrarna ur perspektivet av både variabeltuvinning och klassificering. Variabelutvinning genomförs sekventiellt med olika parameteruppsättningar. Modelluppbyggningsprocessen utförs för varje parameteruppsättning ur variabelutvinningen för att möjliggöra identifiering av vilka parametrar som fungerade bäst under klassificeringen. När de optimala parametrarna för variabelutvinningen funnits, kan parametrarna för maskininlärningsmodellerna vidarejusteras för att förhoppningsvis kunna ytterligare höja klassificeringsnoggrannheterna. Detta kommer att leda till de sk. optimala maskininlärningsmodellerna. Studiens andra steg är att analysera och rapportera resultaten av de optimala maskininlärningsmodellerna.

Datasetet i denna studie har 57 kolumner och 500 rader, där varje rad består av låtdata för en låt. Kolumnerna 1-4 består av deskriptiv textdata (dvs. genre, artistnamn, låtnamn och albumnamn), medan kolumnerna 5-56 är de aggregerade lågnivåvariablerna. Dessa variabler har erhållits genom att ta medelvärdet och variansen för de tjugo kepstral-koefficienterna i Mel-frekvens, spektrala centroiden, spektrala bandbredden, spektral roll-offet, det kvadratiska medelvärdet, nollgenomgången och det dynamiska tempot. Kolumn 57 innehåller information angående saknade värden från variabelutvinningen och används för integritetssjäl. Inga saknade värden hittades dock, vilket innebär att alla 500 låtar användes i analysen. Slutsatsen var ett balanserat dataset med 100 observationer för varje subgenre.

Samplingsfrekvens, signallängd och offset är de tre parametrar som justerades under variabelutvinningen. Samplingsfrekvensen kontrollerar kvaliteten av signal genom att bestämma hur många sampel som mäts under en sekund av en signal. Diskretisering av en signal via sampling är obligatoriskt, eftersom en signal är en oändlig och oräknelig mängd siffror. Signallängden bestämmer i sin tur hur många sekunder av en låt som bör beaktas, medan offsetet möjliggör att ett givet antal sekunder från början av signalen utelämnas. Ifall inte annat anges, användes standardparametrarna för Librosa v0.6.2.

Enligt Nyquist-Shannon-samplingsteoremet bör samplingfrekvensen vara dubbelt så stor som den frekvens som önskas produceras. Eftersom den mänskliga uppfattningen av ljud är ungefär mellan 20 Hz och 20000 Hz vore en ideal samplingsfrekvens åtminstone 40000 Hz. Den vanliga samplingsfrekvensen för CD-skivor är 44100 Hz, medan digitala ljudfiler i t.ex WAV- eller MP3-format kan ha arbiträra samplingsfrekvenser. Att använda sig av olika samplingsfrekvenser för olika ljudfiler skulle leda till inkonsistenser i variabelutvinningen. Lyckligtvis möjliggör samplingsfrekvenskonversion att denna parameter kan hållas lika för alla ljudfiler. Som standard använder sig Librosa av samplingsfrekvensen 22050 Hz, eftersom detta försnabbar processen. Dessutom har det visat sig att en stor del av betydelsefull data ligger under just denna nivå. Samplingsfrekvensen sattes således först till 22050 Hz.

För att undvika att icke-relevant spektral energi (t.ex tysta eller icke-musikaliska artistiska intron i form av atmosfärljud) vid början av låtar kunde påverka klassificeringsprocessen, bestämdes att ett offset på 30 sekunder skulle användas vid början av varje låt. Eftersom minimilängden för varje låt är 3 minuter, bestämdes att signallängden skulle vara 150 sekunder (dvs. 2,5 minuter). Detta är maximalvärdet för att kunna utvinna så mycket

information ur en signal som möjligt, samt kunna bevara en konsistent signallängd och dessutom kunna använda det planerade offsetet på 30 sekunder.

Härnäst applicerades STFT över samtliga ljudfiler. Under denna fas bestämdes fönsterstorleken och hopplängden. Fönsterstorleken är längden av ramen för varje STFT, som appliceras över en ljudfil. Som standard är fönsterstorleken 2048. Hopplängden i sin tur är längden av ramen mellan varje enskild STFT. Ifall hopplängden skulle läggas till 2048, skulle ingen överlappning ske mellan ramarna. Istället är en passlig hopplängd ungefär en fjärdedel av fönsterstorleken (dvs. 2048/4=512). Fönsterstorleken och hopplängden sattes således permanent till 2048 och 512, respektivt.

En signal som transformerats via STFT resulterar i en n-dimensionell vektor. Denna vektor transformerades i sin tur till ett mel-frekvens-spektrogram, men den ursprungliga n-dimensionella vektorn bevarades även. Ur den n-dimensionella vektorn utvanns de tidsdomänrelaterade variablerna, vilka är nollgenomgången och det dynamiska tempot. Ur mel-frekvens-spektrogrammet i sin tur utvanns de tjugo kepstral-koefficienterna i Mel-frekvens, den spektrala centroiden, den spektrala bandbredden, det spektrala roll-offet och det kvadratiska medelvärdet. Slutligen togs medelvärdet och variansen för samtliga variabler, vilket resulterade i de 52 aggregerade variablerna.

Z-scores valdes som normaliseringsmetod för de aggregerade variablerna. Stratifierad K-Fold CV med 10 grupper valdes som valideringsmetod, vilket innebar att varje grupp bestod av 500/10=50 observationer. Stratifiering i sin tur innebär att klassbalans rådde inom varje grupp. Samtliga sju maskininlärningsmodeller tränades med testparametrar på träningsdata och evaluerades sedan på testdata. Processen utfördes separat tio gånger, så att varje indelad grupp funktionerade som testgrupp en gång. Sedan togs medelvärdet av de tio klassificeringsnoggrannheterna för att få ett representativt medelvärde för klassificeringsnoggrannheten för samtliga modeller.

Från den första iterationen med ett samplingsvärde på 22050 Hz var medelvärdet av klassificeringsnoggrannheten för de tre bästa modellerna 56,5 %. Efter att ha provat standardvärdet på 22050 Hz höjdes samplingsfrekvensen till 44100 Hz och 33075 Hz, men inga förbättrade resultat uppnåddes. Tvärtom var medelnoggrannheten en aning lägre, nämligen 54,3 % och 51,8 %, respektivt. Förvånansvärt nog uppnåddes en högre noggrannhet (59,0 %) genom att istället halvera samplingsvärdet till 11025 Hz. Efter detta testades det

ännu lägre samplingsvärdet 5513 Hz, vilket ledde till försämrade resultat (57,0 %). Slutligen höjdes samplingsvärdena ytterligare från 11025 Hz till 15000 Hz och 19000 Hz. Resultaten av denna fas visade att det optimala samplingsvärdet ligger ungefär vid 15000 Hz, vilket gav en medelnoggrannhet på 60,3 % för de tre bästa modellerna.

Eftersom en signallängd på 150 sekunder och ett offset på 30 sekunder hittills använts, var nästa steg att pröva om högre noggrannheter kunde uppnås genom en förändring av dessa två parametrar. I det första försöket utfördes variabelutvinning på hela signalen, dvs. signallängden lades till ljudfilens totala längd och offsetet slopades. I det andra försöket lades signallängden också till den totala längden, men istället för ett manuellt offset inkluderades en automatisk signaltrimning vid början och slutet av signalen. Denna automatiska trimning skär bort tysta sektioner från början och slutet av signalen, vars inklusion kan orsaka skevhet bland de aggregerade variablerna. Den automatiska trimningen beaktar dock inte icke-relevant spektral aktivitet i form av artistiska intron eller outron.

Även om den längsta låten i datasetet är ungefär 13 minuter lång, medan medianlängden är lite under 5 minuter, visade det sig att variabelutvinningen från de 150 sekunder långa klippen ledde till de bästa resultaten. Detta innebär att 150 sekunder var mer än tillräckligt för att generalisera låten som helhet. Som tidigare spekulerats visade det sig att ett offset på 30 sekunder kan vara lönsamt att inkludera, eftersom denna metod gav några procentenheter högre noggrannhetsresultat.

Efter att de bästa parametrarna för variabelutvinning samlats, var det följande steget att prova ifall en annan normaliseringsmetod kunde användas för att höja resultaten. En annan populär normalisationsmetod, min-max, vägdes mot den hittills använda z-scores metoden, men resultaten var ungefär 1,5 procentenheter lägre. Detta kan bero på att min-max har en tendens att påverkas kraftigare av uteliggare.

Efter detta applicerades den inbäddade ExtraTrees-metoden för variabelselektion. Denna metod påminner om Random Forests-inlärningsalgoritmen, där ett stort antal beslutsträd skapas för att kunna identifiera vilka variabler som var av största värde under de simulerade klassindelningarna. Denna metod tillåter rangordning av variablerna utifrån hur viktiga de är. Enligt den resulterade rangordningen utfördes klassificeringsprocessen genom att använda de 40, 30, 20 och 10 bästa variablerna. De bästa resultaten uppnåddes dock med den ursprungliga mängden, dvs. med alla 52 variabler.

Slutligen tränades maskininlärningsmodellerna med ytterligare mer utrymme för finjustering av modellparametrarna. Hittills hade GridSearchCV använts, vilket är en omfattande sökning över de modellparametrarna som användaren specificerat. Genom att förlänga detta parameterutrymme förväntades bättre klassificeringsnoggrannheter kunna uppnås. Resultaten visade dock att de bästa resultaten redan hade uppnåtts med modellernas testparametrar. Detta innebär att testparametrarna var tillräckligt utförliga för att kunna fånga de underliggande sambandena i de utvunna variablerna. Stödvektormaskinerna med en radial basfunktion visade sig vara den mest effektiva inlärningsalgoritmen med en medelnoggrannhet på 62,8 %. Det andra bästa resultatet uppnåddes med Random Forests (60 %) och det tredje bästa resultatet med det artificiella neuronnätet (58,0 %).

Efter att de optimala modellerna anskaffats kunde rapporterna och analysen av de optimala modellerna utföras. Givet de optimala maskininlärningsmodellerna kan det vara av intresse att veta vilka variabler som var mest värdefulla. Resultaten av ExtraTrees visade att de fem kepstral-koefficienterna i Mel-frekvens var av största värde. Resultaten visade också att varians är en mer effektiv aggregationsmetod än medelvärde.

För att kunna uppnå en visuell förståelse av de egentliga observationerna användes oövervakad inlärning i form av principalkomponentsanalys. Visualiseringen visade att subgenrerna överlappar varandra till relativt hög grad, fastän grumliga klustergrupperingar kunde urskiljas. Detta indikerar på att mer sofistikerade variabler krävs för att kunna fånga sambandet med högre noggrannhet. Träningshistorien för det artificiella neuronnätet visar i sin tur att neuronnätet relativt snabbt lär sig sambanden. Även om neuronnätet lär sig sambandet för träningsdata med hundraprocentig noggrannhet, stagnerar resultaten snabbt, då inlärningen istället evalueras med hjälp av testdata. Detta tyder på överträning och det stärker teorin om att ytterligare fler betydelsefulla variabler eller mer data krävs för att högre klassificeringsnoggrannheter ska kunna uppnås.

För att uppnå insikter om vilka genrer som ofta förväxlades med varandra användes procentuella förväxlingsmatriser. Black metal var den lättaste subgenren att klassificera, vilket kan bero på dess generellt diskanta ljudvärld. Oftast blandades denna subgenre med death metal, vilket kan bero på att den rutna sångstilen i melodisk death metal kan ha liknelser med den som används i black metal. Heavy metal var den näst lättaste subgenren att klassificera. Mest ofta förväxlades heavy metal med folk metal och thrash metal, vilka är just de subgenrerna vars utveckling heavy metal har inspirerat. Med samma resonemang

förväxlades death metal oftast med thrash metal och vice versa, vilket är rimligt då death metal utvecklades från thrash metal. Förvånansvärt nog förväxlades folk metal inte oftast med black metal, vilken är en av de subgenrerna den påstås ha utvecklats från. Folk metal förväxlades istället oftast med heavy metal, vilket är den andra subgenren den påstås ha utvecklats från. En orsak till denna förväxling kan vara att den spektrala energin av de akustiska delarna av folk metal driver ned den spektrala energin. Detta i sin tur kan vara en orsak till att folk metal förväxlades med just heavy metal, eftersom den spektrala energin i de tre andra subgenrerna är generellt sett för hög för att lätt kunna förväxlas med.

6.8 Diskussion

Det är skäligt att relativt nya forskningsfält kan drabbas av negativa faktorer som förvränger forskingsresultaten. Detta är speciellt förståeligt då multidimensionella forskningsfält så som MIR och maskininlärning korsar varandra. För vetenskapliga syften kan resultaten från tidigare forskning kräva återevaluering. Denna studie lyckades återevaluera en stor del av den hittills utförda AMGC-relaterade forskningen inom MIR, som drabbats av integritetsproblem. Detta uppnåddes genom ett skräddarsytt dataset, som fokuserade på heavy metal-musik och som samtidigt tog i beaktande alla faktorer som tidigare kunnat leda till integritetsproblem. De relativt lägre resultaten ur denna studie indikerar att resultaten hittills varit överoptimistiska.

För att i verkliga livet kunna skapa värde med ett AMGC-system på ett successivt sätt krävs mer data av hög kvalité, så att bristfälligt klassificerad musikdata inom stora databaser ska kunna klassificeras på ett realistiskt och säkert sätt. Komplexiteten av polyfoniska signaler, samt musikgenrernas dynamiska natur gör i sin tur klassificeringsprocessen utmanande. Lösningen kunde ligga i faktorer så som mer sofistikerad variabelutvinning, aggregationsmetoder, signaltransformationsmetoder eller maskininlärningsmodeller. Fastän uppgiften är utmanande bör successiva AMGC-system eftersträvas, då de har stort värdeskapandepotential för strömningsföretag, eftersom de möjliggör förbättrade metoder för personaliserad marknadsföring.

De tre forskningsfrågorna för denna studie var:

1. Kan maskininlärning användas till urskiljning av subgenrer av heavy metal-musik?

2. Vilken är den mest effektiva lärningsalgoritmen för urskiljandet av subgenrerna?

3. Vilka insikter kan härledas från processen?

För att besvara den första forskningsfrågan gällande om maskininlärning kan användas för urskiljning av subgenrer av heavy metal-musik är svaret subjektivt. En modell utan kunskap skulle statistiskt sett med fem subgenrer uppnå klassificeringsnoggrannheter kring 20 %. Den största klassificeringsnoggrannheten på 62,8 % tyder på att AMGC-systemet fungerar relativt beundransvärt.

Gällande den andra forskningsfrågan om vilken som var den mest effektiva inlärningsalgoritmen är svaret stödvektormaskiner med en radial basfunktion, vilket gav ett noggrannhetsmedelvärde på 62,8 % över de tio testgrupperna. Den andra bästa inlärningsalgoritmen var Random Forests (60,0 %) och den tredje bästa det artificiella neuronnätet (58,0 %). De mer simpla modellerna som tillämpade inlärningsalgoritmerna Gaussian Naïve Bayes och k-Nearest Neighbors uppnådde också relativt goda resultat på 54,4 % och 53,6 %, respektivt.

Den tredje forskningsfrågan gällande vilka insikter som kan härledas från processen har flera svar. För det första tyder resultaten på att samplingsfrekvensen har stor inverkan på resultaten, möjligtvis större än vad som tidigare har diskuterats. En för hög samplingsfrekvens kan icke-intuitivt leda till försämrade resultat och det vanliga värdet på 22050 Hz är dessutom inte alltid det optimala. För det andra kan låtar generaliseras från relativt korta låtsnuttar, vilket kan vara av intresse då AMGC-system byggs från väldigt stora databaser. Detta framkom av parameterjusteringsprocessen under variabelutvinningen, då signallängden på 150 sekunder visade sig leda till högre resultat än den maximala signallängden, samt den maximala trimmade signallängden. För det tredje kan ett manuellt offset vara gynnsamt vid konstruering av AMGC-system, vilket framkom ur parameterjusteringsprocessen för variabelutvinningen. Detta kan bero på offsetets förmåga att exkludera potentiell icke-relevant spektral energi av olika sorter från början av låtar. För det fjärde borde varians som aggregationsmetod föredras över medelvärde, vilket framkom av den inbäddade ExtraTrees-metoden för variabelselektion. För det femte är de högre kepstral-koefficienterna i Mel-frekvens av största värdet vid

urskiljning av heavy metal-subgenrer, vilket också framkom ur de relativa viktighetsgraderna från ExtraTrees-metoden.

Den möjligtvis viktigaste insikten som kan härledas från studien är att mer högkvalitétsdata i form av fler betydelsefulla variabler bör inkluderas i framtida forskning, så att klassificeringsnoggranheten ytterligare ska kunna höjas. Eftersom lyriken bland subgenrerna tenderar att ha sin egen nisch, kunde detta användas till nytta med hjälp av språkteknologi (eng. Natural language processing). Exempelvis tenderar black metal lägga fokus på ämnen så som religion och natur, medan thrash metal viktar mot samhällskritik och krig. Folk metal fokuserar i sin tur oftast på mytologi och historia, medan lyriken i death metal ofta är grafiskt våldsam.

Utöver de nuvarande variablerna kunde variabelutvinning med en så kallad ”bag-of-words”- eller ”word embedding”-modell därför inkluderas för förbättrade resultat. En annan potentiell förbättringsmetod kunde vara inklusion av tag-relaterade data från andra källor. Slutligen kan förbättringar möjligtvis också uppnås med ännu mer komplexa djupinlärningsmodeller. Dessa djupinlärningsmodeller kunde dessutom samarbeta med varandra genom att låta dem rösta för vilken den slutliga prediktionen bör vara. Exempelvis kunde en modell använda sig av sekventiella data istället för aggregerade data, medan en annan modell kunde direkt matas med spektrogram.

References

Ali, M. & Siddiqui, Z. (2017). Automatic Music Genres Classification using Machine Learning. *International Journal of Advanced Computer Science and Applications,* **8**(8), 337-344.

Alpaydin, E. (2010). *Introduction to Machine Learning: Second Edition.* Cambridge, Massachusetts: The MIT Press.

Bahuleyan, H. (2018). *Music Genre Classification using Machine Learning Techniques*.

Barbedo, J. G., & Lopes, A. (2007). Automatic Genre Classification of Musical Signals. In: *EURASIP Journal on Advances in Signal Processing*, 2007, 1-12.

Bennett, K., Parrado-Hernández, E. (2006). The Interplay of Optimization and Machine Learning Research. *Journal of Machine Learning Research*, **7**, 1265-1281.

Bishop, C. (2006). *Pattern Recognition and Machine Learning.* Springer-Verlag New York.

Boutsinas B., Tsekouronas I.X. (2004). Splitting Data in Decision Trees Using the New False-Positives Criterion. In: Vouros G.A., Panayiotopoulos T. (eds) *Methods and Applications of Artificial Intelligence*. SETN 2004. Lecture Notes in Computer Science, 3025. Springer, Berlin, Heidelberg.

Bowar, C. (1.7.2017). What Is Melodic Death Metal, *ThoughtCo*. (Read 5.5.2019). URL: https://www.thoughtco.com/what-is-melodic-death-metal-1756186

Breiman, L. (2001). Random Forests. *Machine Learning*, **45**, 5-32.

Brownlee, J. (2016). *Deep Learning with Python: Develop Deep Learning Models on Theano and TensorFlow using Keras (edition: v1.7).*

Chathuranga, D., Jayaratne, L. (2013). Automatic Music Genre Classification of Audio Signals with Machine Learning Approaches. In: GSTF International Journal on Computing (JoC), **3**(2).

Criminisi, A., Shotton, J., Konukoglu, E. (2012), Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning, *Foundations and Trends® in Computer Graphics and Vision*, **7**, 81-227.

Cutler A., Cutler D.R., Stevens J.R. (2012) Random Forests. In: Zhang C., Ma Y. (eds) *Ensemble Machine Learning*. Springer, Boston, MA.

Davy, M. (2006). An Introduction to Statistical Signal Processing. In: Klapuri, A., Davy, M. (eds) *Signal Processing Methods for Music Transcription,* 21-52. Springer, NY.

Delgado, R. (16.11.2018). How big data has changed the music industry, *Innovation Enterprise*. (Read 27.1.2019). URL: https://channels.theinnovationenterprise.com/articles/how-big-data-has-changed-the-music-industry

Dixon, M., Klabjan, D., Hoon Bang, J (2016). Classification-based Financial Market Prediction using Deep Neural Networks. *Algorithmic Finance*, **6**(3-4), 67-77.

Domingos, P. (2012). A Few Useful Things to Know about Machine Learning, *Communications of the ACM*, **55**(10), 78-87. ACM New York, NY, USA.

D'Souza (21.3.2018). A Quick Guide to Boosting in ML, *Medium*. (Read 8.3.2019). URL: https://medium.com/greyatom/a-quick-guide-to-boosting-in-ml-acf7c1585cb5

Ferreira, A. & Figueiredo, M. (2012). Boosting Algorithms: A Review of Methods, Theory and Applications. In: Zhang C., Ma Y. (eds) *Ensemble Machine Learning*. Springer, Boston, MA.

Flexer, A. (2006). Statistical Evaluation of Music Information Retrieval Experiments. *Journal of New Music Research,* **35**, 113-120.

Geurts, P., Ernst, D., Wehenkel, L. (2006). In: *Machine Learning*, **63**(1), 3-42.

Gibson, J., Van Segbroeck, M., Narayanan, S. (2014). Comparing Time-Frequency Representations for Directional Derivative Features. *Proceedings of the Annual Conference of the International Speech Communication Association*, Interspeech.

Hady, M., Schwenker, F. (2013). Semi-supervised Learning. In: Bianchini, M., Maggini, M., Jain, L. (eds) *Handbook on Neural Information Processing.* Springer-Verlag Berlin Heidenberg.

Hagen, R. (2011). Musical Style, Ideology, and Mythology in Black Metal. In: Wallach, J., Berger, H., Greene, P. (eds) *Metal Rules the Globe,* 180-199. Duke University Press.

Harrington, P. (2012). *Machine Learning in Action.* Manning Publications Co. Greenwich, CT, USA.

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction (Second Edition)*. Springer-Verlag New York.

Heidenreich, H. (5.12.2018). What are the types of machine learning, *Towards Data Science*. (Read: 4.7.2019). URL: https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f

International Data Corporation (2017), *Data Age 2025: The Evolution of Data to Life-Critical.*

Jain, R. (2.2.2017). Introduction to Naive Bayes Classification Algorithm in Python and R, *HackerEarth*. (Read 8.3.2019). URL: https://www.hackerearth.com/blog/machine-learning/introduction-naive-bayes-algorithm-codes-python-r/

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning: With Applications in R.* Springer-Verlag New York.

Johnson, B. (28.8.2018). Artificial Intelligence (AI) – Top Use Cases and Technologies Used Today, *Medium*. (Read 2.2.2019). URL: https://medium.com/@Brian.johnson_62680/artificial-intelligence-ai-top-use-cases-and-technologies-used-today-3c22e1a63e78

Järvinen, P., Järvinen, A. (2004). *On Research Methods*, Opinpajan kirja.

Kahn-Harris, K. (2007). *Extreme Metal: Music and Culture on the Edge.* Berg Publishers.

Klapuri, A. (2006). An Introduction to Statistical Signal Processing. In: Klapuri, A., Davy, M. (eds) *Signal Processing Methods for Music Transcription,* 3-17. Springer, NY.

Kober, J., Bagnell, J., Peters, J. (2013). Reinforcement Learning in Robotics: A Survey. In: *The International Journal of Robotics Research*, **32**(11), 1238-1274.

Kotsiantis, S., Kanellopoulos, D., Pintelas, P. E. (2006). Data Preprocessing for Supervised Learning. In: *International Journal of Computer Science*, **1**(1), 111-117.

Kotsiantis, S. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, **31**, 249-268.

Lampropoulos, A. & Tsihrintzis, G. (2015). *Machine Learning Paradigms: Applications in Recommender Systems.* Springer International Publishing Switzerland.

Lerch, A. (2012). *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics.* Wiley-IEEE Press.

Librosa (17.7.2019). *Why resample on load?* (Read: 26.9.2019). URL: https://librosa.github.io/blog/2019/07/17/resample-on-load/

Marin, E. (26.3.2018). AI-driven data could be the music industry's best marketing instrument, *VentureBeat.* (Read 23.1.2019). URL: https://venturebeat.com/2018/03/26/ai-driven-data-could-be-the-music-industrys-best-marketing-instrument/

Marjenin, P. (2014). *The Metal Folk: The Impact of Music and Culture on Folk Metal and the Music of Korpiklaani.* Master's Thesis in Arts. Kent State University.

Marsland, S. (2015). *Machine Learning: An Algorithmic Perspective (Second Edition).* Boca Raton, FL: Taylor & Francis Group.

McClelland, C. (4.12.2017). The Difference Between Artificial Intelligence, Machine Learning and Deep Learning, *Medium*. (Read 21.10.2018). URL: https://medium.com/iotforall/the-difference-between-artificial-intelligence-machine-learning-and-deep-learning-3aa67bff5991

McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., Nieto, O. (2015). Librosa: Audio and Music Signal Analysis in Python. In: *14th Python in Science Conference.*

Mulder, D.G.J. (2014). *Automatic Classification of Heavy Metal Music.* Bachelor's thesis in Mathematics and Computer Science.

Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective.* Cambridge, Massachusetts: The MIT Press.

Müller, M., Ellis, D., Klapuri, A., Richard, G. (2011). Signal Processing for Music Analysis. In: *IEEE Journal of Selected Topics in Signal Processing*, **5**(6), 1088-1110.

Müller, M. (2015). *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications.* Springer International Publishing Switzerland.

Nasridinov, A. & Park, Y. (2014). A Study on Music Genre Recognition and Classification Techniques. In: *International Journal of Multimedia and Ubiquitous Engineering*, **9**(4), 31-42. SERSC.

Olsen, B. (2008). *I am the Black Wizards: Multiplicity, Mysticism and Identity in Black Metal Music and Culture*. Master's Thesis in Arts. Bowling Green State University.

Panagakis, Y., Kotropoulos, C., Arce, G. (2009). Music genre classification via sparse representations of auditory temporal modulations. European Signal Processing Conference.

Pillsbury, G. (2006). *Damage Incorporated: Metallica and the Production of Musical Identity.* Taylor & Francis Group, NY.

Plate, T. (2000). Accuracy versus interpretability in flexible modeling: implementing a tradeoff using Gaussian process models. *Behaviormetrika*, **26**(1).

Polikar R. (2012) Ensemble Learning. In: Zhang C., Ma Y. (eds) *Ensemble Machine Learning*. Springer, Boston, MA.

Purcell, N. (2003). *Death Metal Music: The Passion and Politics of a Subculture*. McFarland & Company, Inc.

Qin, Z., Liu, W., Wan, T. (2013). A Bag-of-Tones Model with MFCC Features for Musical Genre Classification. In: Motoda et al. (eds) *Advanced Data Mining and Applications: 9th International Conference*, ADMA 2013, 564-575.

Richert, W. & Coelho, L.P. (2013). *Building Machine Learning Systems with Python.* Birmingham, UK: Packt Publishing Ltd.

Rosner, A. & Kostek, B. (2018). Automatic Music Genre Classification Based on Musical Instrument Track Separation. In: *Journal of Intelligent Information Systems*, **50**(2), 363-384. Springer US.

Sarkar, D., Bali, R., Sharma, T. (2018). *Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems.* Apress.

Scaringella, N., Zoia, G., Mlynek, D. (2006). Automatic Genre Classification of Music Content: A Survey. In: *IEEE Signal Processing Magazine*, **23**(2), 133-141.

Schapire R.E. (2003) The Boosting Approach to Machine Learning: An Overview. In: Denison D.D., Hansen M.H., Holmes C.C., Mallick B., Yu B. (eds) *Nonlinear Estimation and Classification.* Lecture Notes in Statistics, **171**. Springer, New York, NY.

Serizel, R., Bisot, V., Essid, S., Richard, G. (2017). Acoustic Features for Environmental Sound Analysis. In: Virtanen, T., Plumbley, M., Ellis, D. *Computational Analysis of Sound Scenes and Events.* Springer, 71-101.

Shalev-Shwartz, S. & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press.

Sharma, S. (23.9.2017). Epoch vs Batch Size vs Iterations, *Towards Data Science*. (Read 7.3.2019). URL: https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9

Shekokar, S., Mali, M. (2013). A brief survey of a DCT-based Speech Enhancement System. In: *International Journal of Scientific & Engineering Research*, **4**(2).

Shete, D., Patil, S. (2014). Zero crossing rate and Energy of the Speech Signal of Devanagari Script. In: *IOSR Journal of VLSI and Signal Processing*, **4**(1), 1-5.

Sturm, B. (2012). A Survey of Evaluation in Music Genre Recognition. In: A. Nürnberger et al. (eds.) *Adaptive Multimedia Retrieval 2012*, LNCS 8382, 29–66. Springer International Publishing Switzerland 2014.

Sturm, B. (2013). *The GTZAN Dataset: Its contents, its faults, their effects on evaluation, and its future use.*

The Economist Intelligence Unit (2015). *Big data evolution: Forging new corporate capabilities for the long term.*

Tsatsishvili, V. (2011). *Automatic Subgenre Classification of Heavy Metal Music.* Master's thesis in Music, Mind & Technology. University of Jyväskylä.

Tzanetakis, G., & Cook, P. (2002). Musical Genre Classification of Audio Signals. In: *IEEE Transactions on Speech and Audio Processing*, **10**(5), 293-302.

VanderPlas, J. (2017). *Python Data Science Handbook: Essential Tools for Working with Data*. Sebastopol, California: O'Reilly Media.

Wallace, W. (1969). *Sociological Theory*. Chicago: Aldine.

Weinstein, D. (2000). *Heavy Metal: The Music and its Culture, Revised Edition.* DaCapo Press.

Weinstein, D. (2014). Pagan Metal. In: Weston, D., Bennett, A. *Pop Pagans: Paganism and Popular Music.* Routledge.

Zhang, M. (9.9.2018). AI's Growing Role in Music Composition, *Medium.* (Read 23.1.2019). URL: https://medium.com/syncedreview/ais-growing-role-in-musical-composition-ec105417899

Zhang, S., Zhang, C., Yang, Q. (2003). Data preparation for data mining. In: Trappl, R. (2003). *Applied Artificial Intelligence*, **17**, 375-381. Taylor & Francis.

Zheng, A. & Casari, A. (2018). *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists.* O'Reilly Media, Sebastopol, CA.

Appendix

BLACK METAL

artist	song	album
...and Oceans	Trollfan	The Dynamic Gallery of Thoughts
1349	Sculptor of Flesh	Hellfire
Alghazanth	Soulquake	The Polarity Axiom
Antimateria	Kun Aukeaa Mysteerit Kuoleman	Valo aikojen takaa
Aoratos	Thresher	Gods Without Name
Aorlhac	La Révolte des Tuchins	L'esprit des vents
Arcturus	The Deep Is the Skies	Aspera Hiems Symfonia (2002)
Asagraum	Transformation	Potestas Magicum Diaboli
Aura Noir	Conqueror	Black Thrash Attack
Azaghal	Vihasta Ja Veritöistä	Nemesis
Barathrum	Regent of Damnation	Saatana
Batushka	Yekteniya 1	Litourgiya
Behexen	Fist of the Satanist	By the Blessing of Satan
Besatt	Hellstorm	Hellstorm
Bezmir	Arrival	Void
Black Altar	Pulse Ov The Universe	Suicidal Salvation
Burzum	Dunkelheit	Filosofem
Carach Angren	The Sighting is a Portent of Doom	Death Came Through a Phantom Ship
Carpathian Forest	Doomed to Walk the Earth As Slaves of the Living Dead	Morbid Fascination of Death
Catamenia	Kingdom of Legions	Eternal Winter's Prophecy
Cirith Gorgor	The Black Hordes	Cirith Gorgor
Cobalt	Gin	Gin
Cosmic Autumn	Event Horizon	Cosmic Autumn
Cosmic Church	Armolahja	Täyttymys
Cradle of Filth	Cruelty Brought Thee Orchids	Cruelty and the Beast

Dark Fortress	Ylem	Ylem
Dark Funeral	Open The Gates	De Profundis Clamavi Ad Te Domine
Darkthrone	Transilvanian Hunger	Transilvanian Hunger
Deathspell Omega	Abscission	Paracletus
Diabolical Masquerade	Haunted by Horror	Nightwork
Dimmu Borgir	Mourning Palace	Enthrone Darkness Triumphant
Dissection	The Somberlain	The Somberlain
Djevel	Vi Slakter Den Foerste og Den Andre, Den Tredje Lar Vi Gaa Mot Nord	Norske Ritualer
Drudkh	Furrows of Gods	Blood in our Wells
Enepsigos	Pagan Rites	Plague of Plagues
Enthroned	The Apocalypse Manifesto	The Apocalypse Manifesto
Evilfeast	Immerse into Cold Mist	Mysteries of the Nocturnal Forest
Forteresse	Par La Bouche De Mes Canons	Thèmes Pour La Rébellion
Frozen Shadows	Through Fields of Mercilessness	Hantises
Förgjord	Epätoivon Virta	Henkeen Ja Vereen
Gaahls Wyrd	Carving The Voices	GastiR (Ghosts Invited)
Gehenna	Werewolf	WW
Gorgoroth	Carving a Giant	Ad Majorem Sathanas Gloriam
Grafvitnir	Wolf of the Eclipse	Venenum Scorpionis
Graveworm	I Need A Hero	Collateral Defect
Hades	Hecate (Queen of Hades)	...Again Shall Be
Horna	Marraskuussa	Envaatnags Eflos Solf Esgantaavne
Hyperion	Novus Ordo Seclorum	Seraphical Euphony
Immortal	Solarfall	At the Heart of Winter
Inquisition	From Chaos They Came	Bloodshed Across the Empyrean Altar Beyond the Celestial Zenith
Insane Vesper	Seed of Inanna	LayiL
Kampfar	Hat Og Avind	Kvass
Keep of Kalessin	Obliterator	Through Times of War
Kjeld	Skym	Skym
Kvist	Ars Manifestia	For Kunsten Maa Vi Evig Vike

Lord Belial	Hymn of the Ancient Misanthropic Spirit of the Forest	Kiss the Goat
Lustre	Part 1	They Awoke to the Scent of Spring
Magoth	Cosmic Termination	Anti Terrestrial Black Metal
Malist	Uniformity	In the Catacombs of Time
Marduk	Serpent Sermon	Serpent Sermon
Mayhem	Pagan Fears	De Mysteriis Dom Sathanas
Mephorash	Chalice of Thagirion	Chalice of Thagirion
Mgla	Exercises In Futility V	Exercises In Futility
Mörk Gryning	Journey	Tusen år har gått
Mörker	Höstmakter	Höstmakter
Naglfar	The Brimstone Gate	Ex Inferis
Pure Wrath	Colourless Grassland	Ascetic Eventide
Ragnarok	My Refuge in Darkness	Arising Realm
Realm of Wolves	Shores of Nothingness	Shores of Nothingness
Rotting Christ	In Nomine Dei Nostri	Rituals
Sacramentum	Far Away From The Sun	Far Away From The Sun
Saor	Bròn	Forgotten Paths
Sargeist	Empire of Suffering	Let the Devil In
Satanic Warmaster	When Thunders Hail	Fimbulwinter
Satyricon	Mother North	Nemesis Divina
Sear Bliss	Seven Springs	Letters from the Edge
Shining	Neka morgondagen	Halmstad
Skogen	När Solen Bleknar Bort	Skuggorna kallar
Spectral Wound	Woods from Which the Spirits Once so Loudly Howled	Infernal Decadence
Summoning	Long Lost To Where No Pathway Goes	Stronghold
Svadilfare	Hordalands Skimmer	Fortapte Roetter
Svarttjern	For What Blooms Without Lust	Towards the Ultimate
Taake	Nattestid Ser Porten Vid I	Nattestid Ser Porten Vid
Temple of Evil	The Book of Shadows	The 7th Awakening
The Kovenant	Chariots of Thunder	Nexus Polaris

The Ruins Of Beverast	The Clockhand's Groaning Circle	Unlock the Shrine
Theosophy	Call of Ugra	Eastland Tales, Pt. I
Thornium	Beyond Cosmic Borders	Mushroom Clouds and Dusk
Throne of Katarsis	Lysets Endeligt	Helvete, Det Iskalde Mørket
Thy Infernal	Rotting in Hell	Warlords of Hell
Thyrane	Chaotic Profane Phenomena	The Spirit of Rebellion
Tsjuder	Norge	Antiliv
Ulver	I Troldskogen Faren Vild	Bergtatt
Urgehal	Goatcraft Torment	Goatcraft Torment
Vinterland	Wings of Sorrow	Welcome My Last Chapter
Watain	Storm of the Antichrist	Sworn to the Dark
Wiegedood	Parool	De Doden Hebben Het Goed III
Windir	Svartesmeden og lundamyrstrollet	Arntor
Wolves in the Throne Room	Born From The Serpent's Eye	Thrice Woven
Woods of Desolation	The Inevitable End	Torn Beyond Reason

DEATH METAL

artist	**song**	**album**
Akercocke	Disappear	Renaissance in Extremis
Allegaeon	Of Mind and Matrix	Proponent for Sentience
Amon Amarth	Down the Slopes of Death	Versus the World
Arch Enemy	Diva Satanica	Burning Bridges
Arsis	We Are The Nightmare	We Are The Nightmare
At The Gates	Slaughter of the Soul	Slaughter of the Soul
Autopsy	Severed Survival	Severed Survival
Barbarity	Blood on My Hands	The Wish to Bleed
Barren Earth	Our Twilight	Curse of the Red River
Before the Dawn	Phoenix Rising	Rise of the Phoenix
Benediction	Subconscious Terror	Subconscious Terror
Beyond Creation	The Inversion	Algorythm

Bloodbath	Cancer Of The Soul	Nightmares Made Flesh
Bloodred Hourglass	Valkyrie	Where the Oceans Burn
Bloodsoaked	Abssesion	Frost Image
Callenish Circle	Obey Me	Flesh Power Dominion
Cannibal Corpse	Shredded Humans	Eaten Back to Life
Carbonized	Recarbonized	For the Security
Carnation	Chapel of Abhorrence	Chapel of Abhorrence
Carrion	In The End, There Is Only Death	Time to Suffer
Children of Bodom	Lake Bodom	Something Wild
Chronolyth	Revenants	Atrophy
Coffinborn	Enter the Nightmares of Horrors	Beneath the Cemetery
Crematory	Chunks of Flesh	Denial
Dark Lunacy	Aurora	The Diarist
Dark Tranquillity	Through Smudged Lenses	Character
Dawn Of Tears	A Cursed Heritage	Act III The Dying Eve
Death	Crystal Mountain	Symbolic
Decapitated	Winds of Creation	Winds of Creation
Deicide	Dead By Dawn	Deicide
Demigod	Slumber of Sullen Eyes	Slumber of Sullen Eyes
Depravity	Silence of the Centuries	Silence of the Centuries
Devenial Verdict	Elysium	Corpus
Disarmonia Mundi	Guilty Claims	Nebularium
Dismember	Override of the Overture	Like an Everflowing Stream
Dominhate	Immolation Carmen Astri	Emissaries of Morning
Dying Fetus	Destroy the Opposition	Destroy the Opposition
Entombed	Living Dead	Clandestine
Eroded	Thousands Cults of Sterility	Engravings of a Gruesome Epitaph
Eternal Tears Of Sorrow	Midnight Bird	Children of the Dark Waters
Excruciate	Passage of Life	Passage of Life
Fractal Gates	Inertia	Altered State of Consciousness
Grave	Amongst Marble and the Dead	Endless Procession of Souls

Gruesome	Inhumane	Twisted Prayers
Horrendous	The Somber (Desolate Winds)	Sweet Blasphemies
Immolation	When the Jackals Come	Atonement
In Flames	Crawl Through Knives	Come Clarity
In Mourning	A Vow to Conquer the Ocean	The Weight of Oceans
Insomnium	Mortal Share	Above the Weeping World
Izegrim	Victim of Honor	Code of Consequences
Kalisia	Awkward Decision	Cybion
Kalmah	The Groan of Wind	The Black Waltz
Krabathor	Faces Under The Ice	Cool Mortification
Krisiun	Devouring Faith	Scourge Of The Enthroned
Krypts	Open The Crypt	Unending Degradation
Lik	Dr Duschanka	Carnage
Magenta Harvest	...And Then Came The Dust	...And Then Came The Dust
Malevolent Creation	Slaughterhouse	Invidious Dominion
Massacre	Dawn of Eternity	From Beyond
Master	The Parable	The Witchhunt
Mastifal	Cultura Brutal	Cultura Brutal
Medeia	Insectia	Xenosis
Miasmal	Death Mask	Miasmal
Morbid Angel	Immortal Rites	Altars of Madness
Morgoth	Isolated	Cursed
Mors Principium Est	Two Steps Away	The Unborn
Mors Subita	Defeat	Into the Pitch Black
Morta Skuld	Dying Remains	Dying Remains
myGRAIN	Alienation	Signs of Existence
Necrophagist	Only Ash Remains	Epitaph
Necrophobic	Unholy Prophecies	The Nocturnal Silence
Neter	Triumphant March	Idols
Nile	Cast Down The Heretic	Annihilation of the Wicked
Norther	Darkest Time	Dreams of Endless War

Nothgard	The Sinner's Sake	The Sinner's Sake
Noumena	Misanthropolis	Anatomy of Life
Obituary	Chopped In Half	Cause of Death
Obscenity	Out of the Tombs	Where Sinners Bleed
Omnium Gatherum	The Unknowing	Beyond
Orbit Culture	Saw	Redfog
Origin	Manifest Desolate	Omnipresent
Persefone	The Majestic of Gaia	Spiritual Migration
Pestilence	Out of the Body	Consuming Impulse
Purtenance	Black Vision	Member of Immortal Damnation
Quo Vadis	On The Shores Of Ithaka	Day into Night
Scar Symmetry	Morphogenesis	Holographic Universe
Sickening Gore	Obscene Existence	Destructive Reality
Slugathor	Journey into Oblivion	Circle of Death
Soilwork	Witan	Verkligheten
Sotajumala	Paratiisin kutsu	Kuolemanpalvelus
Stench of Decay	Creation of Carnal Lust	Stench of Decay
Suffocation	Liege of Inveracity	Effigy of the Forgotten
The Crown	Cobra Speed Venom	Cobra Speed Venom
The Duskfall	Striving To Have Nothing	Frailty And Source
The Exiled Martyr	Catatonic Misery	Insight
The Nomad	Faceless	Victim of the Evolution
Tomb of Finland	Scattering Ashes	Frozen Beneath
Unleashed	Before the Creation of Time	Where No Life Dwells
Vader	Shadowfear	Impressions in Blood
Vital Remains	Dawn of the Apocalypse	Dawn of the Apocalypse

FOLK METAL

artist	**song**	**album**
Adrian von Ziegler	Metsän Läpi	Lifeclock
Alestorm	Over the Seas	Captain Morgan's Revenge

Alkonost	The Indiscernible Path	The Path We've Never Made
Antti Martikainen	Lords of Iron	Northern Steel
Arkona	Odna	Slovo
Bifröst	RundeUmRunde	Tor in eine neue Welt
Blackguard	This Round's On Me	Profugus Mortis
Bran Barr	Celebration (Son of Nuadh Amhach)	Sidh
Bucovina	Spune tu, Vant	Sub Steele
Crimfall	Wildfire Season	As the Path Unfolds...
Crom	The Stars Will Fall	Vengeance
Cruachan	Blood for the Blood God	Blood for the Blood God
Dalriada	A Dudás	Napisten Hava
Drakum	Around the Oak	Around the Oak
Eiswerk	Der Freiheit Entgegen	Kameraden Des Todes
Eluveitie	Inis Mona	Slania
Ensiferum	Wanderer	Victory Songs
Equilibrium	Blut Im Auge	Sagas
Falkenbach	Eweroun	Asa
Fferyllt	Dance of Druids	Dance of Druids
Finntroll	Fiskarens Fiende	Nattfödd
Finsterforst	Urquell	...zum Tode hin
Folkearth	Skaldic Art	By the Sword of My Father
Folkodia	Thus A Viking Dies	Odes from the Past
Forefather	Miri It Is	Steadfast
Fulka	Put Your Faith In Evil	Fulkuna
Fängörn	The Sword of Discord (The First Sword)	Where the Tales Live On
Grai	Within the Forests	About Native Land
Grimner	Enharjarkvade	Frost Mot Eld
Gwydion	From Hel To Asgard	Horn Triskelion
Gymir	Valkyrie Of Sorrow	The Return of the Raven
Hagbard	Let Us Bring Something For Bards To Sing	Warrior's Legacy
Heidevolk	Saksenland	Walhalla wacht

Heol Telwen	Kan Ar Kern	An Deiz Ruz
Holy Blood	My Fate	Shining Sun
Hromovlad	Lithewa	Ohňa hlad, vody chlad
Huldre	Gennem Marsken	Intet Menneskebarn
Hyubris	Orpheu	Forja
In Extremo	Herr Mannelig	Verehrt und Angespien
Incursed	Homeland	Fimbulwinter
Irminsul	Vigridslätt	Irminsul
Ithilien	Blindfolded	Shaping the Soul
Jonne	Pimeä On Oksan Taitto	Kallohonka
Kalevala	Nagryanuli	Luna I Grosh
Kanseil	Panevìn	Doin Earde
KerecsenSólyom	Feasting Field of Heroes	Aquileia Ostroma
Kivimetsän Druidi	Kristallivuoren Maa	Taival
Korpiklaani	Wooden Pints	Spirit of the Forest
Krampus	Rebirth	Survival of the Fittest
Kroda	Werwolf	Varulven
KromleK	Träskens näve	Kveldrihur
Kylfingar	Kilenc Valkűr	Halhatatlanok
Lagerstein	Drink the Rum	All for Rum & Rum for All
Lumsk	I lytinne två	Åsmund Frægdegjevar
Menhir	Das Hildebrandslied, Teil I	Hildebrandslied
Metsatöll	See on see maa	Karjajuht
Mithotyn	The Old Rover	Gathered Around the Oaken Table
Moonsorrow	Pakanajuhla	Suden Uni
Myrkgrav	Fela Etter'n Far	Trollskau, Skrømt Og Kølabrenning
Mägo de Oz	Fiesta pagana	Finisterra
Månegarm	Hemfärd	Vredens Tid
NightCreepers	Tale of Haste	Alpha
Nomans Land	Torir Scald	Raven Flight
Northland	Where the Heroes Die	Northland

Oak Roots	The Branch of Fate	The Branch of Fate
Odroerir	Heimdall	Götterlieder II
Pagan Reign	Novgorodian Folk Dance	Tverd
Pimeä Metsä	Varangian Odyssey	Legacy of the Heathen North
Rivendell	The Old Walking Song	Farewell, The Last Dawn
SatanaKozel	Bitva	Sun of the Dead
Saurom	Cambia el Mundo	Vida
Silent Stream Of Godless Elegy	Winter Queen	Themes
Skiltron	By Sword and Shield	The Clans Have United
Skyforger	Oh Fog, Oh Dew	Pērkoņkalve
Skálmöld	Gleipnir	Börn Loka
Slartibartfass	St. Cuthbert	Nebelheim
Spellblast	Goblin's Song	Horns of Silence
Svartby	Humus	Riv, Hugg Och Bit
Svartsot	Midsommer	Vældet
The HU	Yuve Yuve Yu	The Gereg
The Privateer	Draft of the Strange	The Goldsteen Lay
Thyrfing	Mjölner	Urkraft
Trelleborg	Metsänhumppa!	Lands of Njord
Troldhaugen	Día Del Chupacabra	Obzkure Anekdotez For Maniakal Massez
TrollfesT	Espen Bin Askeladden	Norwegian Fairytales
Tuatha de Danann	Queen of the Witches	Tuatha de Danann
Turisas	Miklagard Overture	The Varangian Way
Týr	Hold the Heathen Hammer High	By the Light of the Northern Star
Valhalore	Upon the Shores	Upon the Shores
Vallorch	Sylvan Oath	Neverfade
Waylander	Born to the Fight	Once Upon an Era
Wolfarian	Dhá Lasair	Beyond the Ninth Wave
Wolfchant	Revenge	Bloody Tales of Disgraced Lands
Wolfhorde	Unyielding	Towards the Gate of North

Wolfsangel	Of Ye Birch Tree Slain	Widdershins
XIV Dark Centuries	Julenzeit	Jul
Ymyrgar	Hall Of The Slain	The Tale As Far
Znich	Dunaju	Slova Ziamli
Zrymgöll	Wood Morning	Creatures of the Night
Ásmegin	Til Rondefolkets Herskab	Hin Vordende Sod & Sø

HEAVY METAL

artist	**song**	**album**
Accept	Fast as a Shark	Restless and Wild
Acero Letal	Duro Metal	Veloz Invencible (Duro Metal)
Acid	Hooked on Metal	Acid
Alpha Tiger	Long Way of Redemption	Identity
Ambush	Firestorm	Firestorm
Amulet	Bloody Night	The First
Angel Witch	Witching Hour	As Above, So Below
Anvil	Jackhammer	Metal on Metal
Apollo Ra	To Be A Hero	Ra Pariah
Assailant	Power of the Hunter	First Offense
Avenger	Revenge Attack	Killer Elite
Bashful Alley	Running Blind	Running Blind
Battleaxe	Chopper Attack	Power From the Universe
Black Rose	Ridin' Higher	Roxcalibur
Black Sabbath	Iron Man	Paranoid
Bleak House	No Reply	Suspended Animation
Blitzkrieg	Buried Alive	Buried Alive
Cauldron	No Return (In Ruin)	In Ruin
Cloven Hoof	Cloven Hoof	Cloven Hoof
Dark Star	Lady of Mars	Dark Star
Diamond Head	Am I Evil	Ligthning to the Nations
Dio	End of the World	Master of the Moon

Dokken	Kiss of Death	Back for the Attack
Down	On March the Saints	Over the Under
Enchanter	Tomb of the Unknown Soldier	Defenders of the Realm
Enforcer	Mesmerized by Fire	Death by Fire
Eternal Champion	The Cold Sword	The Armor of Ire
Gamma Ray	Man on a Mission	Land of the Free
Halford	Like There's No Tomorrow	Halford IV (Made of Metal)
HammerFall	Always Will Be	Gates of Dalhalla
Heavylution	Burn Out	Children of Hate
Helloween	I'm Alive	Keeper of the Seven Keys, Part I
Hocculta	Warning Games	Warning Games
Hollow Ground	Fight With The Devil	Hollow Ground
Icon	World War	Icon
Iron Maiden	The Trooper	Piece of Mind
Iron Savior	Riding on Fire	Riding on Fire
Jag Panzer	Burning Heart	Age of Mastery
Judas Priest	Painkiller	Painkiller
Kamelot	When the Lights Are Down	The Black Halo
Keel	The Right to Rock	The Right to Rock
King Diamond	Lies	Conspiracy
Kotiteollisuus	Minä Olen	Helvetistä Itään
Krokus	Tokyo Nights	Metal Rendezvous
Leather Heart	Destiny	Comeback
Lord Fist	Road Ravens	Green Eyleen
Loudness	Crazy Nights	Thunder in the East
Lunar Shadow	Metalian	Triumphator
Manowar	Hail and Kill	Kings of Metal
Medieval Steel	To Kill a King	The Dungeon Tapes
Mercyful Fate	Evil	Melissa
Metal Church	Badlands	Blessing in Disguise
Motörhead	Overkill	Overkill

Neuronspoiler	Through Hell We March	Emergence
Overlord SR	Keeper of the Flame	Medieval Metal
Ozzy Osbourne	Crazy Train	Blizzard of Ozz
Persian Risk	Ridin' High	Ridin' High
Powervice	Behold the Hand of Glory	Heavy Metal Killers
Queensrÿche	Guardian	Condition Hüman
Quiet Riot	Cum on Feel the Noize	Metal Health
Rage	Make My Day	Secrets in a Weird World
Ratt	You're In Love	Invasion of Your Privacy
Raven	Don't Need Your Money	Don't Need Your Money
Riot	Thundersteel	Thundersteel
Rocka Rollas	Heavy Metal Kings	The War of Steel Has Begun
Ruler	Mirror of Lies	Rise to Power
Running Wild	Lonewolf	Blazon Stone
Sacral Rage	Master of a Darker Light	Deadly Bits of Iron Fragments
Samson	Riding With the Angels	Shock Tactics
Saracen	Crusader	Heroes, Saints & Fools
Satan	Alone in the Dock	Court in the Act
Savage	Let it Loose	Loose 'n Lethal
Savatage	Hall of the Mountain King	Hall of the Mountain King
Saxon	Thunderbolt	Thunderbolt
Seventh Son	Immortal Hours	Immortal Hours
Sinner	Danger Zone	Danger Zone
Skelator	Agents of Power	Agents of Power
Spellcaster	Power Rising	Under the Spell
Stallion	The Right One	Mounting the World
Steel Horse	In the Storm	In the Storm
Stratovarius	Black Diamond	Fourth Dimension
Striker	Former Glory	Striker
Sumerlands	The Seventh Seal	Sumerlands
Sweet Savage	No Guts, No Glory	Regeneration

Tanagra	Tyranny of Time	None of This Is Real
Tank	Kill	Honour & Blood
Tokyo Blade	Unleash The Beast	Night of the Blade
Traitors Gate	Shoot To Kill	Devil Takes the High Road
Twisted Sister	Love Is for Suckers	Love Is for Suckers
Tyran Pace	Shockwaves	Long Live Metal
Tyrant	We Stay Free	Mean Machine
Tyson Dog	Blood Money	Crimes of Insanity
Universe	Weekend Warrior	Weekend Warrior
Victim	Victim	Power Hungry
Virgin Steele	Victory is Mine	The Marriage of Heaven and Hell Part II
W.A.S.P.	Wild Child	The Last Command
Warfare	This Machine Kills	Pure Filth
Warlord	Child of the Damned	Deliver Us
Witchfinder General	Witchfinder General	Death Penalty
Yngwie Malmsteen	I'll See The Light Tonight	Marching Out

THRASH METAL

artist	**song**	**album**
4ARM	Submission For Liberty	Submission For Liberty
Acid Fury	The Hunt	The Hunt
Acid Reign	Reflections of Truths	The Fear
Algebra	Polymorph	Polymorph
Alkoholizer	Thrash Metal	Drunk or Dead...
Amken	Soul's Crypt	Theater of the Absurd
Amok	Thrash Island	Downhill Without Brakes
Angelus Apatrida	You Are Next	The Call
Annihilator	No Way Out	Feast
Apocalypse	Digital Life	Apocalypse
Artillery	The Challenge	Terror Squad
Assassin	The Last Man	The Upcoming Terror

Atrophy	Preacher, Preacher	Socialized Hate
Attomica	The Chainsaw	Disturbing the Noise
Axegressor	Lead Justice	Last
Breathless	Nuclear Seas	Thrashumancy
Burning Nitrum	High Speed Bangers	Molotov
Bywar	The Last Life	Heretic Signs
Condition Critical	Parasitic Torment	Operational Hazard
Conflagrator	Knowledge (Is Madness)	Knowledge (Is Madness)
Coroner	Sudden Fall	Punishment for Decadence
Cripper	Animal of Prey	Antagonist
Crisix	UItra Thrash	The Menace
Dark Angel	The Burning of Sodom	Darkness Descends
Death Angel	Mistress Of Pain	The U
Deathgeist	Thrash Metal Fire	Thrash Metal Fire
Deathraiser	Terminal Disease	Violent Aggression
Deathrow	The Deathwish	Deception Ignored
Destruction	Unconscious Ruins	Release from Agony
Dust Bolt	Soul Erazor	Awake the Riot
Eruption	Fractured	Tenses Collide
Evil Invaders	As Life Slowly Fades	Feed Me Violence
Evile	Thrasher	Enter the Grave
Exarsis	Skull And Bones	The Human Project
Exhorder	Unforgiven	The Law
Exodus	Til' Death Do Us Part	Pleasures of the Flesh
Exumer	Catatonic	The Raging Tides
Farscape	Killers on the Loose	Killers on the Loose
Forbidden	Chalice of Blood	Forbidden Evil
Fueled by Fire	Thrash Is Back	Spread the Fire
Gama Bomb	Terrorscope	The Terror Tapes
Game Over	No More	Burst Into The Quiet
Gammacide	Victims of Science	Victims of Science

Hatchet	Signals of Infection	Dawn of the End
Havok	Killing Tendencies	Time Is Up
Heathen	Mercy is No Virtue	Victims of Deception
Hellcannon	Harbinger of War	Infected With Violence
Hexen	State of Insurgency	State of Insurgency
Holy Terror	Do Unto Others	Mind Wars
Kreator	Coma of Souls	Coma of Souls
Lazarus A.D.	Revolution	The Onslaught
Lich King	Combat Mosh	Born of the Bomb
Lost Society	Lethal Pleasure	Terror Hungry
Mantic Ritual	One By One	Executioner
Merciless Death	Death Warriors	Realm of Terror
Mortal Sin	Women In Leather	Mayhemic Destruction
Mortillery	F.O.A.D.	Origin of Extinction
Mutant Squad	Mutants Will Rise	Titanomakhia
Mörbid Carnage	Slaughtered	Night Assassins
Nailbomb	Wasting Away	Point Blank
Nervosa	Masked Betrayer	Victim of Yourself
Neurotoxin	Dead That Live	Harvest Your Wrath
Nightbreed	Beyond Inferno	Beyond Inferno
Nuclear	Criminal Solicitation	Jehovirus
Nuclear Assault	Rise From The Ashes	Survive
Nuclear Omnicide	Merciless Butcher	Bringers of Disease
Onslaught	Killing Peace	Killing Peace
Overkill	Live Young, Die Free	Horrorscope
Overthrow	Repressed Hostility	Within Suffering
Pandemia	Suicide Squad	Behind Enemy Lines
Prestige	Angels Cry	Decades of Decay
Project Pain	Primator	Brothers in Blood
Raging Fury	Man Spider	Raging Fury
Razor	High Speed Metal	Malicious Intent

Riffocity	Hail Thy Father	Under a Mourning Sky
Sabbat	Do Dark Horses Dream of Nightmares	Dreamweaver
Sacred Reich	Surf Nicaragua	Surf Nicaragua
Sacrifice	Terror Strikes	Forward to Termination
Sadus	Undead	Chemical Exposure
Shrapnel	Eternal War	The Devastation to Come
Slammer	Tenement Zone	The Work of Idle Hands...
Slayer	Angel of Death	Reign in Blood
Sodom	Agent Orange	Agent Orange
Space Eater	A Thousand Plagues	Passing Through the Fire to Molech
Stone	Get Stoned	Stone
Suicidal Angels	Apokathilosis	Sanctify the Darkness
Tankard	A Girl Called Cerveza	A Girl Called Cerveza
Tantara	Prejudice of Violence	Based on Evil
Testament	Rise Up	Dark Roots of Earth
Thraw	Doomsday Code	Doomsday Code
Tormenter	Absolution	Pulse of Terror
Toxic Holocaust	Nowhere to Run	Conjure and Command
Traitor	Thrash Command	Thrash Command
Vektor	Tetrastructural Minds	Demolition
Viking	White Death	Man of Straw
Vindicator	Thrash and Destroy	There Will Be Blood
Violent Force	Dead City	Malevolent Assault of Tomorrow
Warbringer	Severed Reality	Waking into Nightmares
Warfect	Drone Wars	Exoneration Denied
Weresquatch	Frozen Void	Frozen Void