



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Engenharia Elétrica e de Computação

Vítor Yudi Shinohara

**Explorations on Automatic Music Content  
Labeling using Lyrics**  
**Explorações sobre rotulagem automática de  
conteúdo musical usando letras de músicas**

Campinas

2021

Vítor Yudi Shinohara

# **Explorations on Automatic Music Content Labeling using Lyrics**

## **Explorações sobre rotulagem automática de conteúdo musical usando letras de músicas**

Dissertation presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, in the area of Computer Engineering.

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Engenharia de Computação.

Supervisor: Prof. Dr. Tiago Fernandes Tavares

Este trabalho corresponde à versão final da dissertação/tese defendida pelo aluno Vítor Yudi Shinohara, e orientada pelo Prof. Dr. Tiago Fernandes Tavares.

Campinas

2021

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Rose Meire da Silva - CRB 8/5974

Shinohara, Vítor Yudi, 1997-  
Sh63e Explorations on automatic music content labeling using lyrics / Vítor Yudi Shinohara. – Campinas, SP : [s.n.], 2021.

Orientador: Tiago Fernandes Tavares.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Processamento de linguagem natural (Computação). 2. Aprendizado de máquina. I. Tavares, Tiago Fernandes, 1984-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Explorações sobre rotulagem automática de conteúdo musical usando letras de músicas

**Palavras-chave em inglês:**

Natural language processing (Computer science)

Machine learning

**Área de concentração:** Engenharia de Computação

**Titulação:** Mestre em Engenharia Elétrica

**Banca examinadora:**

Tiago Fernandes Tavares [Orientador]

Romis Ribeiro de Faissol Attux

Rodrigo Hübner

**Data de defesa:** 13-12-2021

**Programa de Pós-Graduação:** Engenharia Elétrica

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0002-4067-8161>

- Currículo Lattes do autor: <http://lattes.cnpq.br/0723692265384229>

## COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato(a): Vítor Yudi Shinohara RA: 229990

Data de defesa: 13 de Dezembro de 2021

Título da Dissertação: “Explorations on Automatic Music Content Labeling using Lyrics”

Prof. Dr. Tiago Fernandes Tavares (Presidente)

Prof. Dr. Romis Ribeiro de Faissol Attux

Prof. Dr. Rodrigo Hübner

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

# Acknowledgements

I would like to thank my advisor, professor Tiago Fernandes Tavares for his support, dedication and patience during the master's degree, contributing immensely to my academic growth. I also thank professors Romis Ribeiro de Faissol Attux and Rodrigo Hübner for their contributions to this work.

I thank my family, especially my parents Lúcio Shinohara and Roseli Ogura Shinohara, for their continued support and for striving for me to have the opportunity to perform this research.

I also thank my friends Matheus Sapia Guerra and Thailon Ferreira Tavares for always supporting me in the most challenging times.

I am very grateful to professor Juliano Henrique Foleiss, for his confidence in referring me to professor Tiago, making this journey possible.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. (Proc. n. 88887.343209/2019-00).

# Resumo

A classificação automática de música consiste em classificar o conteúdo musical em categorias usando algoritmos computacionais. Existem diversas aplicações de classificação automática de música, como a organização de grandes coleções musicais em gêneros, *mood* ou instrumentos utilizados. No entanto, a classificação de músicas é uma tarefa desafiadora e difícil de ser executada com precisão. Existem diversos trabalhos que propõem diferentes métodos de extração de características musicais, a fim de discriminar a música em diferentes categorias. Nesta pesquisa, apresentamos um estudo sobre a classificação automática de músicas utilizando características textuais. Foi utilizado dois extratores de características textuais amplamente usados que se baseiam em frequências de palavras: Bag of words, Term-Frequency - Inverse Document Frequency e um extrator de características semânticas: GloVe. As características baseadas em frequência das palavras e características semânticas foram aplicadas para classificar as letras das músicas em gênero e *mood*, utilizando a Máquina de Vetores de Suporte e Rede Neural Artificial. Além disso, elucidamos o desempenho dos métodos de representação textual usados para classificação através da análise exploratória dos dados. Os modelos de aprendizagem de máquina atingiram 57% de acurácia para a classificação de gênero musical e 74% de acurácia para classificação de *mood*, mas a análise de dados exploratória indica que as letras não são características discriminativas em nenhum dos casos e os resultados são provavelmente devido a algoritmos que exploram características não generalizáveis do conjunto de dados.

**Palavras chave:** processamento de linguagem natural, aprendizagem de máquina.

# Abstract

Automatic music classification consists of classifying music content into categories using computational algorithms. There are a range of applications of the automatic music classification process, such as organization of large musical collections in genres, mood or instruments used. Nonetheless, music classification is a challenging task and hard to perform accurately. There are several works that propose different methods to extract music features in order to discriminate music in different categories. In this research, we present a study on automatic music classification using lyrics features. We utilized two widely used text extractors that relies on word frequency: Bag of words, Term-Frequency - Inverse Document Frequency and a word semantic feature extractor GloVe. We utilized the word frequency and semantic features to classify music lyrics in genre and mood utilizing Support Vector Machine and Artificial Neural Network. Furthermore, we elucidate the performance of the textual representation methods used for classification through exploratory analysis. Machine learning models achieved 57% accuracy for music genre classification and 74% accuracy for mood classification, but exploratory data analysis indicates that lyrics are not discriminative features in either case and results are probably due to algorithms exploiting non-generalizable characteristics of the dataset.

**Keywords:** natural language processing, machine learning.

# List of Figures

Figure 1.1 – Russell’s circumplex emotion model (RUSSELL, 1980) . . . . .	16
Figure 3.1 – Linearly separable and non-linearly separable data. . . . .	20
Figure 3.2 – One dimensional non-linear separable data . . . . .	23
Figure 3.3 – Non-linearly separable dataset becomes linearly separable after mapping the data to a higher dimensional space . . . . .	23
Figure 3.4 – Architecture of a artificial neural network neuron. . . . .	24
Figure 3.5 – Example of a confusion matrix. . . . .	26
Figure 3.6 – Bag of Words feature extractor method to generate feature vectors. . .	29
Figure 3.7 – Example of a two dimensional embedding space. . . . .	30
Figure 4.1 – Methodology used to visualize word embedding vectors . . . . .	41
Figure 4.2 – Topic distribution (topic activation) extraction from rock lyrics. . . .	42
Figure 4.3 – Entropy calculated from topic activations obtained using Latent Dirichlet Allocation . . . . .	42
Figure 5.1 – Visualization of words used in each genre in the word embedding space. We can see that most genres use most words in the embedding space. .	44
Figure 5.2 – Visualization of words used in each mood in the word embedding space	44
Figure 5.3 – Entropy calculated from each genre subset. . . . .	45
Figure 5.4 – Confusion Matrix for genre classification using SVM with TF-IDF features.	46
Figure 5.5 – Confusion Matrix for genre classification using Neural Network with word embedding features. . . . .	47
Figure 5.6 – Confusion Matrix for mood classification using SVM with TF-IDF features.	47
Figure 5.7 – Confusion Matrix for mood classification using Neural Network with word embedding features. . . . .	48
Figure A.1 – Visualization of words used in each genre in the word embedding space. We can see that most genres use most words in the embedding space. .	56
Figure B.1 – Visualization of words used in each mood in the word embedding space	57



# List of Tables

Table 3.1 – Kernel functions (LORENA; CARVALHO, 2007) . . . . .	24
Table 3.2 – Probability of the appearance of the words ice or steam with a context word k. (PENNINGTON; SOCHER; MANNING, 2014) . . . . .	31
Table 3.3 – Probability distribution of words for n topics. . . . .	32
Table 5.1 – Model accuracy and text features used in genre classification experiments	46
Table 5.2 – Model accuracy and text features used in mood classification experiments	46

# List of abbreviations

**AMC** Automatic Music Classification.

**ANN** Artificial Neural Network.

**BoW** Bag of Words.

**GloVe** Global Vectors for Word Representation.

**ISOMAP** Isometric Mapping.

**kNN** k-Nearest Neighbors.

**LDA** Latent Dirichlet Allocation.

**MDS** Multidimensional Scaling.

**MIR** Music Information Retrieval.

**NLP** Natural Language Processing.

**PCA** Principal Component Analysis.

**SVM** Support Vector Machines.

**TF-IDF** Term Frequency - Inverse Document Frequency.

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>13</b>
<b>1.1</b>	<b>Mood</b>	<b>15</b>
<b>1.2</b>	<b>Music Genre</b>	<b>15</b>
<b>1.3</b>	<b>Objectives</b>	<b>16</b>
<b>1.4</b>	<b>Publications</b>	<b>17</b>
<b>2</b>	<b>RELATED WORK</b>	<b>18</b>
<b>3</b>	<b>THEORETICAL BACKGROUND</b>	<b>19</b>
<b>3.1</b>	<b>Machine Learning</b>	<b>19</b>
3.1.1	Support Vector Machines	20
3.1.1.1	Maximum Margin Hyperplane	20
3.1.1.2	Soft Margin Support Vector Machines	22
3.1.1.3	Non linear SVM	22
3.1.2	Artificial Neural Networks	24
3.1.3	Performance Evaluation	25
3.1.3.1	Accuracy	25
3.1.3.2	Confusion Matrix	26
<b>3.2</b>	<b>Natural Language Processing</b>	<b>26</b>
3.2.1	Pre-Processing Techniques	27
3.2.1.1	Special Character Removal and Normalization	27
3.2.1.2	Stopwords Removal	27
3.2.1.3	Lemmatization	28
3.2.2	Textual Features	28
3.2.2.1	Word occurrence	28
3.2.2.2	Word embedding	29
<b>3.3</b>	<b>Topic Analysis</b>	<b>32</b>
3.3.1	Latent Dirichlet Allocation	32
<b>3.4</b>	<b>Dimensionality Reduction</b>	<b>33</b>
3.4.1	ISOMAP	34
<b>4</b>	<b>METHODOLOGY</b>	<b>36</b>
<b>4.1</b>	<b>Dataset</b>	<b>36</b>
4.1.1	Metrolyrics	36
4.1.2	MoodyLyrics	37
4.1.3	Data Preprocessing	37

<b>4.2</b>	<b>Feature Extraction</b>	<b>37</b>
4.2.1	Bag of Words	38
4.2.2	Term Frequency - Inverse Document Frequency	38
4.2.3	GloVe	38
<b>4.3</b>	<b>Music Genre and Mood Classification</b>	<b>39</b>
4.3.1	Support Vector Machine	39
4.3.2	Artificial Neural Networks	40
<b>4.4</b>	<b>Exploratory Analysis</b>	<b>40</b>
4.4.1	Word Embedding Visualization	40
4.4.2	Topic Analysis	41
<b>5</b>	<b>RESULTS AND DISCUSSION</b>	<b>43</b>
<b>5.1</b>	<b>Exploratory Analysis</b>	<b>43</b>
5.1.1	Embedding Visualization	43
5.1.2	Topic Analysis	45
<b>5.2</b>	<b>Music genre and mood classification</b>	<b>45</b>
<b>6</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>49</b>
<b>6.1</b>	<b>Future Work</b>	<b>50</b>
	<b>BIBLIOGRAPHY</b>	<b>51</b>
	<b>ANNEX</b>	<b>55</b>
	<b>ANNEX A – MUSIC GENRE CLASSIFICATION WORD EMBED- DING VISUALIZATION</b>	<b>56</b>
	<b>ANNEX B – MUSIC MOOD CLASSIFICATION WORD EMBED- DING VISUALIZATION</b>	<b>57</b>

# 1 Introduction

With the advancement of technology, the availability and access to multimedia content has become increasingly easy. Physical media such as CDs and vinyls were quickly replaced by digital media. In this way, companies that offer streaming services, such as Spotify, Amazon Music, Tidal and Napster, have become popular, allowing users to access a large amount of multimedia content in a simple way. Consequently, due to the large amount of multimedia content available, index or organize this content automatically becomes a necessity.

One approach to organize large digital media collections is metadata. In the context of music, the metadata stores information regarding the file, such as author, name of the song, year of release, name of the album, among others. However, this information is often not available or not sufficient (MCKINNEY; BREEBAART; (WY, 2003; LI; GUO, 2000).

In this context, research in the Music Information Retrieval (MIR) area seeks to analyze the content of the audio track and extract information and create intelligent systems. Among the various research topics covered in Musical Information Retrieval, we can mention:

- **Automatic Music Transcription:** Convert an acoustic music signal to the corresponding music notation (i.e. song score).
- **Instrument identification:** Identify which instrument classes are present in the audio signal.
- **Cover song detection:** Identify the different versions re-performed or re-recorded of a song by different artists.
- **Music recommendation:** Recommendation of songs that the user are likely to enjoy based on their preferences and tastes.
- **Artist identification:** Identification of the artists or performers of the song.
- **Genre classification:** Classification of the music in a music genre.
- **Mood classification:** Classification of the music based in its mood.

Recently, a problem that has been gaining prominence in the MIR area is the Automatic Music Classification (AMC). The AMC problem can be understood as a classification of songs in taxonomies automatically performed by computers. There

are several advantages to automating music classification using computers. The main advantage is the great speed of music categorization when performed by the computer compared to manual human classification. Humans must listen the music and need seconds  
35 or even minutes to categorize a song. On the other hand, computers extract relevant music features and assign a label almost instantly.

Furthermore, the manual music classification process requires hiring music experts, making the process financially costly. Additionally, each song, ideally, should be analyzed and labeled by more than one specialist, with the objective of alleviating  
40 individual subjectivity in relation to the analyzed music, in addition to increasing the consistency of categorization.

AMC systems can be applied for commercial and academic purposes. As mentioned earlier, one of the applications for automatic music classification is the organization of large music collections. In addition, we can mention that automatic music classification  
45 contributes to the development of music recommendation systems and playlist generation. Playlist generation systems, for example, can use categories of audio tracks, such as musical genre or mood to generate the playlist. However, most AMC applications are not trivial to be developed with precision. Some AMC tasks do not have well-defined concepts or consistent and clear rules for conceptualizing each category, such as genre or mood.

50 To perform automatic music classification, supervised machine learning algorithms are commonly employed. Supervised learning algorithms constructs a theory based on how the inputs of each category are related to the output to effectively predict the output of new inputs (RAMÍREZ; FLORES, 2019).

In these algorithms an input vector is provided, which is a numeric representation of the song, along with its label. The algorithm creates a function which maps  
55 the features of the songs to their respective labels and can be used to infer labels from examples not yet seen.

We can summarize the classification process in three different phases. Firstly, it is necessary to extract relevant information represented by a vector of real numbers to  
60 describe a music. Sequentially, it is required to train a machine learning model to map and identify data patterns, mapping information of a given category to their respective label. Finally, the evaluation and labeling process of unseen examples is based on the mapping built in the training stage (RAMÍREZ; FLORES, 2019).

Music contains different types of content such as lyrics, melody, video clip,  
65 cultural and social data, which we can use to extract information relevant to its categorization. However, it is typical for works addressing AMC to focus on using information extracted only from audio signals (MAYER; RAUBER, 2011).

Melody can often define a song's categorization in terms of music genre. Songs

in the Hip-Hop genre tend to have more bass use. Instruments that compose the music  
70 can indicate the genre of the music as well ([MAYER; NEUMAYER; RAUBER, 2008](#)).

However, some categorizations do not depend solely on the melody. We can cite  
as an example Christmas songs or romance songs. These categories are assigned to songs  
based mainly on the meanings of their lyrics. Lyrics is also a great source of information,  
which can convey thoughts, emotions and feelings related to a particular subject ([Sharma  
75 et al., 2016](#)).

Several studies make use of song lyrics to extract relevant information for music  
classification ([HU; DOWNIE, 2010](#); [KUMAR; RAJPAL; RATHORE, 2018](#); [ZAANEN;  
KANTERS, 2010](#); [YING; DORAISAMY; ABDULLAH, 2012](#)). We can highlight the  
classification in musical genres and mood in the MIR area, as both are essential criteria  
80 for organizing music for users ([YING; DORAISAMY; ABDULLAH, 2012](#)).

## 1.1 Mood

Mood can be conceptualized as an individual's mental state. Several studies  
aim to establish a concept for mood and define which moods exist, however, a universal  
set of moods or emotions has not yet been established.

85 We can distinguish works that aim to categorize emotions into categorical  
emotion models or dimensional emotion models ([Ren; Wu; Jang, 2015](#)). In the categorical  
model, emotions are considered distinct categories like sadness, happiness, boredom.

A well-known work that proposes a categorical emotional model is by Ekman  
([EKMAN, 1984](#)). Ekman categorized emotions into six: expressions, which are anger,  
90 fear, happiness, sadness, disgust and surprise. On the other hand, some works categorize  
emotions in fewer categories, for example sadness and happiness ([WEINER; GRAHAM,  
1984](#)).

On the other hand, there is the dimensional approach. Dimensional models do  
not treat emotions as discrete categories. In this model, emotions are represented through  
95 dimensions. We can mention the "Circumplex model", proposed by Russell, which consists  
of the representation of emotions in two dimensions: arousal, which represents the intensity  
of the emotion and the valence, which defines the polarity of the emotion, illustrated in  
Figure 1.1.

## 1.2 Music Genre

100 Unlike mood classification, music genre classification works do not adopt a set of  
pre-defined labels. Researches in this area are based on the classes available in the dataset,

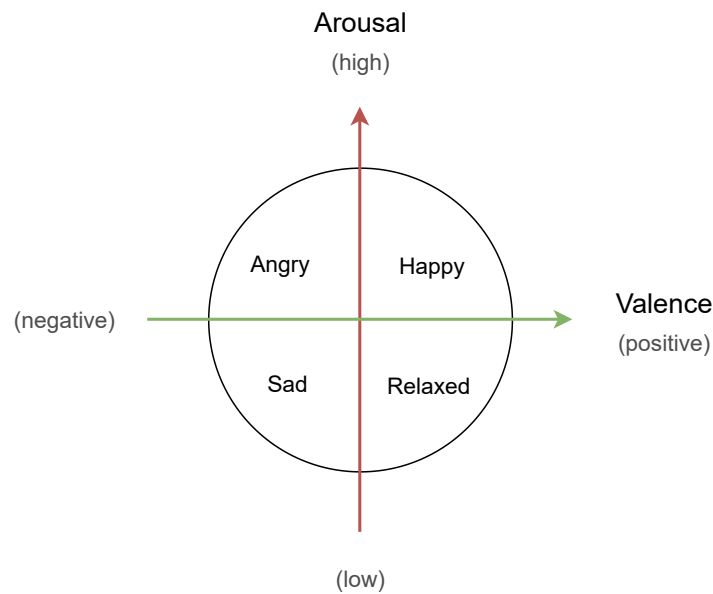


Figure 1.1 – Russell’s circumplex emotion model (RUSSELL, 1980)

commonly varying from 2 to 10 distinct labels (SCARINGELLA; ZOIA; MLYNEK, 2006). In addition, it is important to note that the concept of music genre is subjective. Music genres have no formal definition and well-defined boundaries (PACHET; CAZALY et al., 105 2000). It is possible to observe different criteria for labeling tracks, such as geographic (k-Pop), era in history (Baroque) or technique used (Barbershop) (SCARINGELLA; ZOIA; MLYNEK, 2006). Therefore, the automatic classification of musical genres is a challenging task.

### 1.3 Objectives

110 The primary objective of this research is to evaluate the performance of techniques for automatic classification of music based on textual descriptors. The contribution of this work consists of elucidating the poor performance of machine learning models used to categorize songs in musical genres or mood. Finally, we can summarize the objectives of this research as follows:

- 115
- Further elucidate the reasons underlying performance differences of different lyric-based AMC systems;
  - Relate AMC performance to music genre characteristics;
  - Relate AMC performance to Natural Language Processing (NLP) techniques used to represent text.



## 1.4 Publications

- Shinohara V. Y., Foleiss J. H., Tavares T.F. (2019). **Comparing Meta-Classifiers for Automatic Music Genre Classification.**
- Ogiwara M., Manolovitz B., Shinohara V.Y., Ren G., Tavares T.F. (2021) **Graph-Based Representation, Analysis, and Interpretation of Popular Music Lyrics Using Semantic Embedding Features.** In: Miranda E.R. (eds) Handbook of Artificial Intelligence for Music. Springer, Cham.

## 2 Related Work

This section informs relevant works in the automatic music classification area. We highlight works aimed at categorizing music genres and moods. In addition, we specifically focus on researches that use features derived from song lyrics.

5           The research by Canicatti ([CANICATTI, 2016](#)) only addresses the classification of music genres. The author used 2500 lyrics from the MetroLyrics dataset, distributed in 5 musical genres. The [Bag of Words \(BoW\)](#) model was used to extract information. Subsequently, the [k-Nearest Neighbors \(kNN\)](#), Random Forest and Naïve Bayes classifiers were used. The highest accuracy obtained was 47.35%.

10           Siriket et. al. ([SIRIKET; SA-ING; KHONTHAPAGDEE, 2021](#)) performed a comparison of different machine learning algorithms for mood classification. The author used a dataset composed of 636 song lyrics derived from the Billboard and MoodyLyrics datasets. Song lyrics are unevenly distributed among four moods. Random forest, Decision Tree, Naïve Bayes, Logistic Regression, AdaBoost and XGBoost algorithms were used.  
15 The accuracy obtained was 89.19%, however no evaluation metrics were computed for unbalanced datasets.

          Ying et. al. ([YING; DORAISAMY; ABDULLAH, 2012](#)) performed a study on music genres and mood classification. The authors performed the categorization of 600 song lyrics distributed among 10 music genres and 10 different moods. The lyrics used  
20 in the research were extracted from three distinct websites: LyricWiki, MetroLyrics and Amarok. For the classification process, the [Support Vector Machines \(SVM\)](#), [kNN](#) and Naïve Bayes algorithms were used. The authors achieved an accuracy of 39.94% for music genre classification and 59.67% for mood classification.

          However, the works mentioned above do not seek to elucidate the performance of  
25 machine learning models. Data exploration can show the presence or absence of relationships between textual features and musical categories.

## 3 Theoretical Background

In this chapter, the concepts needed to understand this research will be presented. Initially, we cover the concepts related to machine learning, including the supervised learning models [Support Vector Machines \(SVM\)](#) and [Artificial Neural Network \(ANN\)](#).  
 5 Lastly, we approach the concepts of the [Natural Language Processing \(NLP\)](#) area, formalizing feature extraction algorithms used in this research, such as [Bag of Words \(BoW\)](#), [Term Frequency - Inverse Document Frequency \(TF-IDF\)](#) and [Global Vectors for Word Representation \(GloVe\)](#).

### 3.1 Machine Learning

10 Machine learning is a field of artificial intelligence that studies the development of algorithms and techniques that allow devices to acquire knowledge from large volumes of data. Machine learning techniques are applied in order to improve performance of tasks.

Machine learning algorithms acquire information through inductive inference. Inductive inference allows conclusions to be drawn from a set of examples ([ANGLUIN; SMITH, 1983](#)). We can categorize inductive learning into supervised learning and unsupervised learning.  
 15

Algorithms based on supervised learning require a set of labels ( $Y$ ) for the dataset ( $X$ ) ([ALPAYDIN, 2020](#)). Labels must be chosen based on the problem to be solved ([BISHOP, 2006](#)). Supervised learning systems can be applied to classification tasks and  
 20 regression tasks. In the classification task, the label indicates a categorical value which indicates the class for the example, for example digit recognition where the class can be from 0 to 9. In the regression task, the label is a real value corresponding to the data, for example the price of a property.

The classification or regression task performed by supervised learning algorithms  
 25 can be separated into two different stages. The first stage consists of training the machine learning model. In the training process, a subset of data, called a training set, is presented to the model, along with its labels. In this phase, the model maps the training data to their respective labels. The second stage consists of testing the model, where its generalization performance is measured. The testing process consists of introducing data that the model  
 30 has never been presented to before and obtaining a prediction for that data. Sequentially, the predictions are compared with the true labels from the data and a metric is computed.

A metric widely used to analyze the performance of models is accuracy. Accuracy is calculated by dividing the number of the correct predictions of the model divided by

the total number of predictions.

35        Unsupervised algorithms do not require labels assigned to the data. Unsuper-  
vised algorithms aims to find similarities between the data provided to the algorithm.  
Usually, unsupervised algorithms are used to group data into clusters. The data is assigned  
to the clusters based on their patterns, as measured by a cost function (DUDA; HART;  
STORK, 2001), so similar data is ideally assigned to the same cluster.

40        In this work, we used two widely used supervised machine learning models,  
namely SVM and ANN, for music genre classification and mood classification. The machine  
learning algorithms used in this work will be explained in the following sections.

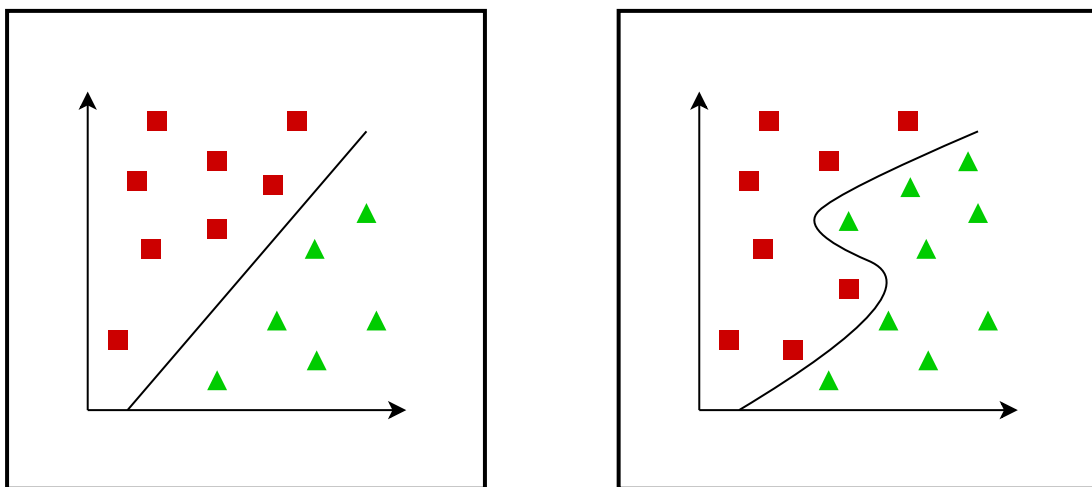
### 3.1.1 Support Vector Machines

45        SVM is a supervised learning algorithm, which is widely used in a variety  
of classification and regression tasks. Results obtained through the application of SVM  
compare or exceed results of models widely used as ANN (HAYKIN; NETWORK, 2004).

From a set of data referring to two classes, named positive and negative, and  
their respective labels, the SVM model seeks to stipulate a decision boundary, named  
hyperplane (BISHOP, 2006). The hyperplane aims to separate the data from the positive  
50    class and the data from the negative class.

#### 3.1.1.1 Maximum Margin Hyperplane

One of the variants of the SVM model is called the Maximum Margin Classifier.  
The Maximum Margin Classifier can only be effectively applied to linear data, which rarely  
represents real-life data. Figure 3.1a illustrates data from two distinct classes, which are  
55    linearly separable. Figure 3.1b illustrates non linearly separable data.



(a) Linearly separable data.

(b) Non-linearly separable data.

Figure 3.1 – Linearly separable and non-linearly separable data.

A hyperplane is considered ideal or maximum margin when it separates data from distinct classes without error. Furthermore, the hyperplane must maximize the distance between data from different categories. The distance from the closest sample of one class to the nearest example of another class is called the separation margin. Thus, to  
 60 find the ideal hyperplane, the separation margin must be maximized.

Considering a linearly separable training set with  $n$  examples, given  $x_i \in X$  and their respective labels  $y_i \in Y$  being  $y_i \in \{-1, 1\}$ , we can assume the existence of a hyperplane that separates the examples of the positive and negative classes. The hyperplane can be formalized by the Equation 3.1. Additionally,  $w$  is normal to the hyperplane,  $\frac{|b|}{||w||}$   
 65 represents the distance from the hyperplane to the origin.

$$f(x) = w.x + b = 0 \quad (3.1)$$

The SVM model with hard margins enforces that there is no training data between the hyperplane  $w.x_i + b = 0$  and its margin  $|w.x_i + b| = 1$ , as illustrated in Equation 3.2.

$$\begin{cases} x_i.w + b \geq +1, & \text{if } y_i = +1 \\ x_i.w + b \leq -1, & \text{if } y_i = -1 \end{cases} \quad (3.2)$$

These can be combined into a single inequality:

$$y_i(x_i.w + b) - 1 \geq 0, \forall i = 1, 2, \dots, n \quad (3.3)$$

70 Sequentially, the distance  $d_i$  between a given  $x_i$  for a hyperplane  $(w, b)$  can be calculated using the Equation 3.4.

$$d_i(w, b, x_i) = \frac{|w.x_i + b|}{||w||} \quad (3.4)$$

Thus, applying the restriction of hard margins expressed in Equation 3.3, the distance between a given  $x_i$  and the hyperplane can be illustrated by:

$$d_i(w, b, x_i) = \frac{1}{||w||} \quad (3.5)$$

Therefore, the lower limit of the distance between a hyperplane and a data of  
 75 a positive  $d_+$  or negative  $d_-$  class is:

$$d_+ = d_- = \frac{1}{||w||} \quad (3.6)$$

Once the distance from the hyperplane to positive or negative class data has been defined, we can define the margin:

$$m = d_+ + d_- = \frac{2}{\|w\|} \quad (3.7)$$

Therefore, to maximize the margin that separates data from distinct classes, minimizing the value of  $\|w\|$  is necessary. The  $\|w\|$  minimization problem can be described  
80 as:

$$\text{Maximize } \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j (x_i \cdot x_j) \quad (3.8)$$

### 3.1.1.2 Soft Margin Support Vector Machines

The constraint imposed by the hard margin SVM model limits its application to data that are linearly separable. As discussed earlier, there are a few scenarios where data falls into this category. One factor we can mention is that in most cases, there is the  
85 presence of outliers or noise in the dataset. The Soft Margin extension allows some data to violate the restriction imposed in Equation 3.3.

This is done by inserting slack variables in the constraint imposed in the SVM model with hard margins. Equation X illustrates the restriction imposed by the Soft Margin model, with  $\xi_i$  being the slack variables, for  $i = 1, 2, \dots, n$ .

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i = 1, \dots, n \quad (3.9)$$

90 The variable  $\xi_i$  represents a training error. Therefore, the sum of  $\xi_i$  is an upper bound for training errors. With the introduction of the slack variable  $\xi$ , the presence of training data between the hyperplanes  $H_1$  and  $H_2$  becomes possible, in addition to tolerating classification errors. Thus, it is also necessary to change the objective function to be minimized.

$$\frac{1}{2} \|w\|^2 + C \left( \sum_{i=1}^n \xi_i \right) \quad (3.10)$$

95 The constant C must be defined and represents the penalty for training errors. A high value of C results in a higher error penalty.

### 3.1.1.3 Non linear SVM

With the introduction of the slack variables in the soft margin SVM, it becomes effective when classifying datasets approximately linear or linear, due to the tolerance of

100 noise and outliers. On the other hand, there are many cases where the model will not be efficient. Figure 3.2 illustrates nonlinear data of only one dimension. Using the data illustrated with the soft margins extension will result in many misclassifications.



Figure 3.2 – One dimensional non-linear separable data

For classification of non-linear data by SVM, a training set is mapped to a new space of greater dimension, which is called feature space. In a high-dimensional space, there is a high probability that the data is separable. However, the mapping or transformation must be linear and the feature space must have sufficiently high dimensionality.

Figure 3.3 illustrates a single-dimensional nonlinearly separable dataset. By transforming the data in one-dimensional space to two-dimensional space using a  $\phi$  function, the data becomes linearly separable.

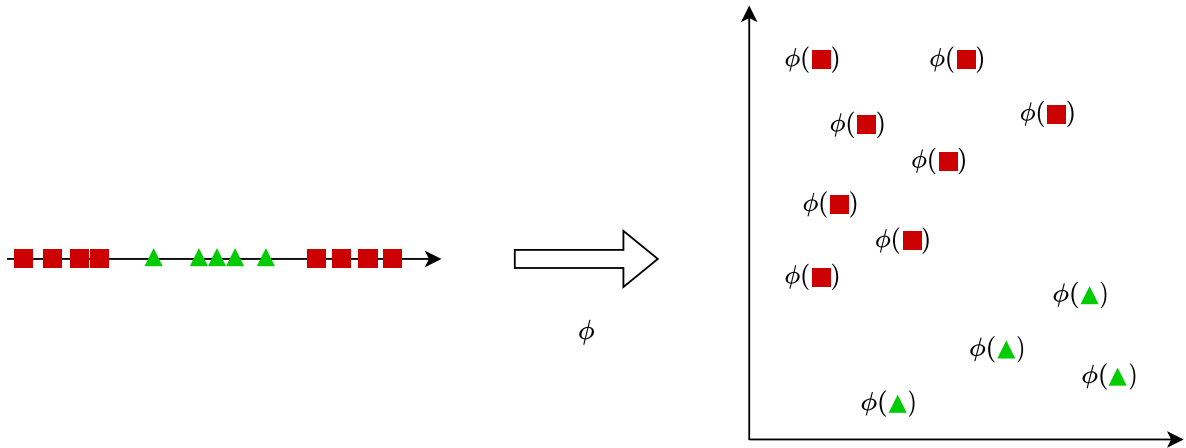


Figure 3.3 – Non-linearly separable dataset becomes linearly separable after mapping the data to a higher dimensional space

110 Thus, the optimization function described in Equation 3.11 can be defined as:

$$\text{Maximize } \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j (\phi(x_i) \cdot \phi(x_j)) \quad (3.11)$$

In real scenarios, the high dimensionality of the feature space obtained by mapping  $\phi$  can result in a high computational cost for calculating  $\phi(x_i) \cdot \phi(x_j)$ . From this computational problem, kernels are employed. Kernels allow the dot product of data belonging to the feature space to be computed without knowing the  $\phi$  mapping, in addition to using the data in the original space. The kernel function  $K$  can be interpreted as follows:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \quad (3.12)$$

Some commonly used kernel functions are described in Table 3.1

Kernel Type	Kernel Function	Params
Gaussian / RBF	$\exp(-\sigma  x_i - x_j  ^2)$	$\sigma$
Polynomial	$(\delta(x_i \cdot x_j) + \kappa)^d$	$\delta, \kappa$ and $d$
Sigmoid	$\tanh(\delta(x_i \cdot x_j) + \kappa)$	$\delta$ and $\kappa$

Table 3.1 – Kernel functions (LORENA; CARVALHO, 2007)

### 3.1.2 Artificial Neural Networks

In the last decade, ANN have become popular for clustering, classification and regression tasks (ABIODUN et al., 2018). Several researches use ANN, due to the great power of pattern recognition (ABIODUN et al., 2018).

The functioning of an ANN is based on processing units, called neurons. Neurons are responsible for outputting an value based on input values. The neuron output is calculated through a linear combination of the input values and sequentially, submitted to an activation function. The activation function aims to make the network non-linear, allowing the model solving more complex problems (FAUSETT, 1994).

The ANN architecture consists of sets of neurons, organized in layers. Normally, neurons are connected to neurons in the sequential layer through unidirectional weighted edges. Edges are responsible for transmitting values between neurons, and the transmitted values are weighted by the value of the edge (FAUSETT, 1994).

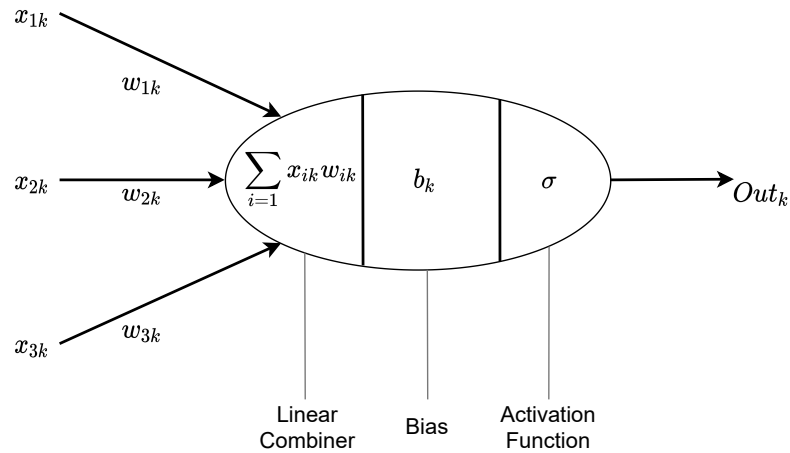


Figure 3.4 – Architecture of a artificial neural network neuron.

Figure 3.4 illustrates the structure of the artificial neuron (perceptron)  $k$  of an ANN. The neuron is connected by 3 edges with the weights  $w_{1k}$ ,  $w_{2k}$  and  $w_{3k}$ . These edges are transmitting the signals  $x_{1k}$ ,  $x_{2k}$  and  $x_{3k}$  by the previous layer neurons. The neuron



input value illustrated in the figure can be formalized as:

$$Out_k = \sigma\left(\sum_{i=1}^n x_{ik}w_{ik} + b_k\right) \quad (3.13)$$

Initially, the weights and bias values are initialized with random values. The  
 135 adjustment of weight and bias values is performed in the neural network training process,  
 and it can be divided in two different processes: forward propagation and back propagation.  
 In the forward propagation process, the data introduced in the network with their respective  
 labels is passed between the neurons of each layer, up to the output layer. After propagating  
 the values through the network, the result emitted by the output layer is compared with the  
 140 expected result through an error function. In the back propagation, the error is propagated  
 from the output layer to the input layer. Then, the values of the weights and bias are  
 modified in order to reduce the back-propagated error (KRÖSE et al., 1993).

### 3.1.3 Performance Evaluation

The performance evaluation of a machine learning model is an essential step.  
 145 The performance evaluation aims to measure the generalizability of the algorithm. That  
 is, measuring the classification capacity of examples not yet introduced to the model.

To evaluate the model performance, it is common to divide the database into  
 three sets, called training set, test set and validation set. Once the model is trained using  
 the training set, the evaluation of its performance is made using the data present in the  
 150 test and validation set. It is important to note that the data in the training set must not  
 be present in the test set or in the validation set.

There are different metrics to measure the generalizability of a model. We will  
 discuss two metrics that are widely used in the literature in the following sections.

#### 3.1.3.1 Accuracy

155 Accuracy is a widely used metric to measure the performance of a machine  
 learning model. The accuracy of a model can be calculated by dividing the number of  
 correct predictions  $Y_{true}$  by the total number of performed predictions  $Y_{pred}$ .

$$Accuracy = \frac{Y_{true}}{Y_{pred}} \quad (3.14)$$

Accuracy tends to work well in cases where the machine learning model has  
 been trained on a balanced dataset, that is, each class contains the same amount of  
 160 examples.

On the other hand, when using an unbalanced dataset, accuracy can be an indicator of erroneous performance. For example, consider a database composed of two classes, with 98% of the examples belonging to class A and 2% belonging to class B. If the model assigns all examples to class A, the accuracy will be 98 %.

### 165 3.1.3.2 Confusion Matrix

The confusion matrix is a widely used method to illustrate the performance of machine learning models used to classify examples into two or more classes. However, a disadvantage of this metric is the need for human interpretation.

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

Figure 3.5 – Example of a confusion matrix.

The confusion matrix informs true-positives, true-negatives, false-positives, and  
 170 false-negatives predictions. From a scenario where the classifier distinguishes examples between positive and negative classes, we can interpret the metrics as follows:

- True-positive: The classifier labeled the data as positive, and the data belongs to the positive class.
- True-negative: The classifier labeled the data as negative, and the data belongs to  
 175 the negative class.
- False-positive: The classifier labeled the data as positive, but the data belongs to the negative class.
- False-negative: The classifier labeled the data as negative, but the data belongs to the positive class.

## 180 3.2 Natural Language Processing

NLP is a subfield of Artificial Intelligence that aims to study and develop techniques that enable machines to understand natural language, being speech and text. This is

done through a combination of machine learning techniques, statistics and computational linguistics. Among the [NLP](#) applications, we can mention:

- 185 • **Information Retrieval:** Information retrieval involves returning a set of documents in response to a user query, for example search engines.
- **Chatbots:** Chatbots are computational algorithms capable of analyzing textual or oral data to generate an appropriate response. It allows users to ask questions like “How are you?” and generate an answer according to what was asked. Chatbots are  
190 commonly used in customers services and information acquisition ([LALWANI et al., 2018](#)).
- **Automatic Text Summarization:** Text summarization consists of capturing relevant information compiled from one or more documents and producing a new document that contains a portion of the information from the original texts ([MAY-  
195 BURY, 1999](#)).
- **Machine Translation:** It consists of converting natural language from one language to another, maintaining its meaning, without human assistance ([MIDDI; RAJU; HARRIS, 2019](#)).

### 3.2.1 Pre-Processing Techniques

200 The text pre-processing consists of applying text processing techniques to minimize language distortions. Pre-processing techniques make the input data for the subsequent classification process more consistent, contributing to increase the classifier’s accuracy ([UYSAL; GUNAL, 2014](#)). In this work, we made the treatment of special characters, removed stopwords and applied the stemming process.

#### 205 3.2.1.1 Special Character Removal and Normalization

Special characters can generate ambiguities in the feature extraction stage. For example, the “cat” and “cat!” tokens are treated differently, even though they represent the same information. The normalization of words to lowercase letters also aims to remove ambiguities, since “Cat” and “cat” will be treated differently, although they represent the  
210 same word.

#### 3.2.1.2 Stopwords Removal

Another widely used pre-processing technique is remove stopwords. We can notice that some words have a great frequency of use in texts. These words are usually prepositions, articles, conjunctions or pronouns, such as “of”, “is”, “are”, “that”, “this”.

215 These words are known as stopwords. Stopwords can negatively influence the result of a text classification process, in addition to increasing the quantity of terms analyzed and consequently, the computational cost.

### 3.2.1.3 Lemmatization

Lemmatization is a method in the area of [NLP](#) that aims to reduce inflected words in the language to its base form, called lemma ([TOMAN; TESAR; JEZEK, 2006](#)). An example is to process the inflected words “walked” and “walking” for the base form “walk”. Consequently, the word vocabulary is reduced, since different inflected words are represented by only one base word. In our experiments, we used the WordNet Lemmatizer<sup>1</sup> algorithm to reduce inflected words to its base form.

## 225 3.2.2 Textual Features

In the context of machine learning, features are characteristics or attributes that describe and represent data. The quality of the features has a great impact on the algorithms performance. In the area of [NLP](#), the features vectors, which are usually a set of real numbers, are derived from textual data. The process of mapping text to numeric  
230 vectors is called feature extraction.

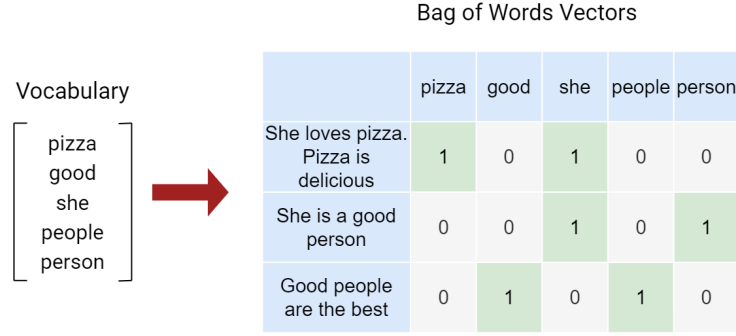
### 3.2.2.1 Word occurrence

It is common for research in the area of [NLP](#) to use feature extractors according to the word occurrence. One of the most widely used and simple feature extractors is the [BoW](#) ([HARRIS, 1954](#)). This approach relies on describing the occurrence of words in a document. It involves choose a set of words to compose a vocabulary of size  $N$ . Then, just  
235 create a word histogram for each document. The histogram has  $N$  positions, where each position indicates the number of appearances of a certain word present in the vocabulary, or its absence (0). In the case of the binary variant, the histogram only compute the presence (1) or absence (0) of the specific word. Figure [3.6](#) illustrates a vocabulary containing 5  
240 words and the binary vectors of each document.

More robust methods consider the importance of words in a document based on the frequency in which they are used, as is the case with [TF-IDF](#) ([SALTON; BUCKLEY, 1988](#)). In this method, each word in a document receives a weight or relevance. The relevance of the word increases with based on its frequency. However, the importance of  
245 the word is balanced by its frequency in the set of documents, called corpus. In conclusion, a word that is common in all documents, has a low relevance. Conversely, a word that has a high frequency in only one document, but does not have a high frequency in the corpus, has greater relevance.

---

<sup>1</sup> <http://nltk.org/>



		Bag of Words Vectors				
Vocabulary		pizza	good	she	people	person
<div> <p>pizza</p> <p>good</p> <p>she</p> <p>people</p> <p>person</p> </div>	She loves pizza. Pizza is delicious	1	0	1	0	0
	She is a good person	0	0	1	0	1
	Good people are the best	0	1	0	1	0

Figure 3.6 – Bag of Words feature extractor method to generate feature vectors.

In this method, the frequency of the term  $t$  in document  $d$  is represented by  $tf(t, d)$ . The inverse document frequency (idf) can be calculated dividing the number of documents  $N$  in a corpus  $C$  by the number of documents that contain the term and applying the logarithm, as represented in Equation 3.15.

$$idf(t, d) = \log \left( \frac{N}{count(d \in C : t \in d)} \right) \quad (3.15)$$

Finally, the **TF-IDF** can be calculated multiplying the term frequency times the inverse document frequency, illustrated at Equation 3.16, resulting in a metric of a term's relevance for a particular document.

$$tf\ idf(t, d) = tf(t, d) \times idf(t, d) \quad (3.16)$$

It is important to mention that both **BoW** and **TF-IDF** do not consider information about the order or position of the words in a sentence. These methods are only concerned in the presence of words in the document. Also, the feature vector of these methods are related to the number of words in the vocabulary. As the vocabulary size increases, the presence of more sparse feature vectors is common. Sparse vectors require more memory and have a higher computational cost for machine learning algorithms.

### 3.2.2.2 Word embedding

Another method of textual representation widely used in **NLP** resesarches is known as word embedding. Word embedding is a technique which maps words to a real space, named embedding space. Words are mapped to the embedding space based on its semantic and syntactic value (WANG et al., 2019; OGIHARA et al., 2018).

A limitation present in the methods of textual representation based on word occurrence is the feature vectors dimensionality. The dimensionality of the vectors is related to the vocabulary size of these methods. In case of very large vocabularies, the classifiers execution time is compromised.

Unlike the textual representation methods through the occurrence of words, the representation of the word embedding is done using fixed size low-dimensional vectors of real numbers. The vectors represent the position of each word in an embedding space, where words with similar meanings and syntax have similar vectors representations (WANG  
 275 et al., 2019), resulting in being close in the embedding space, illustrated in Figure 3.7.

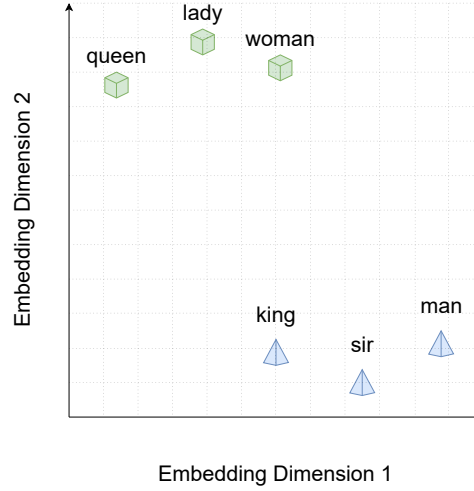


Figure 3.7 – Example of a two dimensional embedding space.

Vector representations of the words “king” and “man” are mapped close together in the embedding space, since they have similar semantic value. For the same reason, the words “woman” and “man” are mapped apart.

There are different algorithms and methods for generating the word embedding  
 280 vectors. A widely used method is GloVe (PENNINGTON; SOCHER; MANNING, 2014).

GloVe is based on the suggestion that words in the same context have a correlated semantic value. First, the method aims to find relationships between words in a sentence using a co-occurrence matrix  $X$ . The co-occurrence matrix contains the information in the cell  $X_{ij}$  of how frequent a word  $w_i$  appears in the context of the word  
 285  $w_j$  in the corpus. We can formalize the probability  $P_{ij}$  of the word  $i$  occurring in the context of the word  $w_j$  as follows:

$$P_{ij} = X_{ij} / \sum_{k=0}^n X_{ik} \quad (3.17)$$

In order to clarify the semantic relationship with the proximity of the appearance of words, Pennington et al. created an example based on the words “ice” and “steam”. Table 3.2 illustrates the probabilities  $P(i|k)$  and  $P(j|k)$  of observing the words “ice” or  
 290 “steam” together with another word  $k$ .

From the words  $i = ice$ ,  $j = steam$  and the word  $k$ , which represents the context, we can observe the variations in probabilities increasing or decreasing according to the

Probability and Ration	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k ice)$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$
$P(k steam)$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$
$P(k ice) / P(k steam)$	8.9	$8.5 \times 10^{-2}$	1.36	0.96

Table 3.2 – Probability of the appearance of the words ice or steam with a context word  $k$ . (PENNINGTON; SOCHER; MANNING, 2014)

semantic relation of the words  $i$ ,  $j$  and  $k$ .

The ratio  $P(i|k)/P(j|k)$  has a high value when the context word has similar  
 295 semantic value compared with the word  $i$ . On the other hand, the same ratio has a low  
 value when word  $k$  has a similar value to word  $j$ . Finally, the value of  $P(i|k)/P(j|k)$   
 approaches 1 when the words have very similar meanings ( $k = water$ ) or very different  
 meanings ( $k = fashion$ ).

Since the co-occurrence matrix captures the semantic relationship between  
 300 three words, the authors use the ratio  $P(i|k)/P(j|k)$  to generate the embedding vectors.  
 This concept can be formalized as illustrated in the following equation.

$$F(w_i, w_j, \tilde{w}_k) = \frac{P(i|k)}{P(j|k)} \quad (3.18)$$

The equation illustrates the optimization problem of the GloVe model. The  
 $F$  function must be applied to the embedding vectors of the words  $i$ ,  $j$  and  $k$ . The result  
 of the  $F$  function must be close to the ratio  $P(i|k)/P(j|k)$ . The authors define the cost  
 305 function of the model, illustrated in the following equation.

$$J = \sum_{i,j=1}^V f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2 \quad (3.19)$$

$V$  represents the vocabulary size,  $w_i$  and  $b_i$  represent the vector and the bias  
 of the word  $i$  respectively,  $\tilde{w}_j$  and  $\tilde{b}_j$  illustrate the vector and the bias of the word  $j$ .  $X$  is  
 the co-occurrence matrix and  $f$  represents a weighting function. The weighting function  
 sets a threshold so that terms that frequently co-occur do not influence the cost. Among  
 310 the possible  $f$  functions, the authors used the following function, where the parameter  
 values are chosen empirically, where  $X_{max} = 100$  and  $a = 3/4$ .

$$f(x) = \begin{cases} (x/x_{max})^a & \text{if } x < x_{max}, \\ 1 & \text{otherwise} \end{cases} \quad (3.20)$$

Finally, through the gradient descent optimization algorithm and calculation  
 of the cost function derivatives with respect to the parameters, the embedding vectors are  
 rectified until their state of convergence (RODRÍGUEZ, 2017).

### 3.3 Topic Analysis

The topic analysis technique is widely used in [NLP](#) works to extract information from documents through unsupervised machine learning algorithms. This technique aims to extract the main subjects, named topics, addressed by the documents of a corpus. There are different topic modeling algorithms and the most used are Latent Semantic Analysis (LSA) ([DUMAIS, 2004](#)), Probabilistic Latent Semantic Analysis (PLSA) ([HOFMANN, 2013](#)) and [Latent Dirichlet Allocation \(LDA\)](#) ([BLEI; NG; JORDAN, 2003](#)).

#### 3.3.1 Latent Dirichlet Allocation

[LDA](#) is a generative probabilistic model proposed for text collections ([BLEI; NG; JORDAN, 2003](#)), which specifies the generation of documents through latent variables. In this context, the latent variables can be understood as topics. The observable variables are the words that compose the documents in a corpus.

The generative model [LDA](#) considers that documents are composed of a distribution of topics. Topics are a probability distribution of a set of words present in a corpus, as illustrated in [Table 3.3](#).

	Word 1	Word 2	Word 3	...	Word k
<b>Topic 1</b>	0.06	0.13	0.33	...	0.09
<b>Topic 2</b>	0.26	0.04	0.13	...	0.11
<b>Topic 3</b>	0.16	0.09	0.12	...	0.41
...	...	...	...	...	...
<b>Topic n</b>	0.07	0.32	0.06	...	0.27

Table 3.3 – Probability distribution of words for n topics.

In order to formalize the [LDA](#) generative process, we will define the following notations:

- $\Theta_i$  : Probability distribution of topics given a document  $d_i$ ;
- $\Phi_j$  : Probability distribution of words of one topic  $z_j$ ;
- $d$  : Document

We can formalize the generation of the document  $d_i$  as samples of the variables  $\Theta$  and  $\Phi$ .  $\Theta_i$  represents the distribution of topics for a given document and  $\Phi_j$  represents the probability distribution of vocabulary words given a topic  $j$ . Each term in a document comes from the sampling of a word through the distribution  $\Phi_j$ . Sequentially, the topic  $j$  was selected by sampling the distribution  $\Theta_i$ .



340 The inverse process of generating documents is to identify the composition of topics that were responsible for generating a particular document (FALEIROS; LOPES, 2016). For this process, statistical inference methods are used, such as the Gibbs Sampler (GEMAN; GEMAN, 1984) and variational inference.

## 3.4 Dimensionality Reduction

345 Textual data feature extraction methods usually use a large number of dimensions in their feature vectors. This is due to the need to map a word to each dimension of the vector, as in the case of the BoW algorithm. However, the computational cost of using high-dimensional feature vectors with a machine learning model is high. In this scenario, dimensionality reduction techniques are applied to reduce the number of dimensions in  
350 the feature space.

In addition, dimensionality reduction can be applied in order to prevent problems caused by the large number of dimensions, such as the curse of dimensionality phenomenon (DONOHO et al., 2000).

Dimensionality reduction is also applied to remove redundant samples present  
355 in the dataset (LIU; MOTODA, 2012). Redundant data can be formalized as entries or attributes in the dataset that do not provide additional information, for example an attribute with the same value for all data or an attribute that can be calculated through other attributes of the same example (LIU; MOTODA, 2012).

The removal of redundant data leads to a better performance of machine  
360 learning models, in addition to contributing to a reduction in computational cost for the training and classification process.

In addition, dimensionality reduction techniques are also applied to allow data visualization, making it possible to identify the data structure (CUNNINGHAM; GHAHRAMANI, 2015).

365 Dimensionality reduction can be formalized as a transformation of high-dimensional data into low-dimensional data that corresponds to the original dimension of the data, preserving the most relevant information (MAATEN et al., 2009). We can distinguish dimensionality reduction methods into two categories: feature selection and feature extraction.

370 Dimensionality reduction algorithms based on feature selection, select a subset of features that better discriminate the classes present in the dataset (NASREEN, 2014). The selection of the subset can be done in several ways, among them, we can mention two widely used methods: forward selection and backward elimination.

- **Forward selection** - This iterative method consists of creating an empty subset of features. At each iteration, the attribute that results in the greatest performance gain from the machine learning model is added to the subset (SCHENK; KAISER; RIGOLL, ).
- **Backward elimination** - In this iterative method, initially the subset of features is composed of all attributes. At each iteration, an attribute is removed based on its impact on the machine learning model metric (NAKARIYAKUL; CASASENT, 2009).

On the other hand, the feature extraction process aims to derive new features through linear or non-linear combinations of the original data. Mathematical combination is done with the intention of reducing the dimension of the original data to a smaller dimension, compressing the relevant information and removing redundant data. Among the dimensionality reduction algorithms through feature extraction, we can mention the **Principal Component Analysis (PCA)** (JOLLIFFE, 2005) and **Isometric Mapping (ISOMAP)** (TENENBAUM; SILVA; LANGFORD, 2000).

### 3.4.1 ISOMAP

The **ISOMAP** algorithm is a non-linear dimensionality reduction method that maps data from a high-dimensional space into a low-dimensional space, preserving the original data structure. The dimensionality reduction is performed by maintaining the distance of the original data in the new reduced dimension space. Therefore, data that were close in the original space remains close in the reduced dimension space (LI et al., 2006).

The dimensionality reduction by the **ISOMAP** algorithm is based on the construction of a graph. Each example in the dataset is represented as a vertex. From this, the shortest path between each pair of vertices is computed. Sequentially, through all the distances between the points, the algorithm aims to find a low-dimensional mapping that preserves the distances.

First, you must build a graph  $G$  from the dataset, where a vertex will represent each example. If two vertices are connected in  $G$ , the edge receives the weight of the Euclidean distance  $D(x_i, x_j)$  between the examples.

The assignment of edges connecting two distinct vertices in the graph is commonly done through the **k-Nearest Neighbors (kNN)** algorithm (COVER; HART, 1967). The **kNN** algorithm calculates the Euclidean distance from given example  $x_i$  to other examples  $x_j$ . From this, the  $K$  examples closest to  $x_i$  are selected. In this way, the

edges between the vertex which represents  $x_i$  are created, connecting to  $X_j$ , weighted by its Euclidean distance.

410           From the definition of vertices and edges of the graph  $G$ , the minimum distance of the vertex  $x_i$  is computed in relation to all other vertices. The minimum distance between one vertex and another is calculated using the Dijkstra algorithm. The minimum distance between the vertices  $x_i$  and  $x_j$  is represented as  $D_{ij}$ .

              From the matrix  $D$ , a set of points in the lesser dimension space must be found,  
415 preserving the given distances through [Multidimensional Scaling \(MDS\)](#).

## 4 Methodology

To investigate the great difficulty in classifying song lyrics in music genres and mood due to the subjectivity of these labels, we propose methods to evaluate and investigate the performance of machine learning models assigned to solve this task.

5 This chapter presents the steps to perform the analysis and classification of song lyrics in mood or musical genres. We present the steps of data pre-processing, training, and performance evaluation of the classifiers used in this research.

We can categorize the experiments in exploratory analysis of song lyrics and classification of data into music genre or mood. The exploratory analysis, in particular,  
10 was performed to investigate the data sets and verify the existence or absence of common characteristics between different musical genres or moods.

In addition to exploratory analysis, we evaluated the performance of two different machine learning models applied to the classification of musical genre and mood, using textual features extracted from song lyrics.

### 15 4.1 Dataset

One of the fundamental steps in developing an automatic music content labeling system is to obtain the dataset. Ideally, a dataset should represent a subset of the real world. Also, the dataset must have “ground truth” labels. However, there is no “ground truth” for labeling music content in genres or moods, due to its subjectivity and lack of  
20 formal rules defining the limits of each category.

In the following sections, two datasets widely used for the tasks of automatic music classification, which were used in this research, are described.

#### 4.1.1 Metrolyrics

In our experiments, we used a subset of the MetroLyrics dataset for the genre  
25 classification task. Metrolyrics dataset is composed of 380.000 songs with lyrics from the genres Rock, Pop, Hip-Hop, Metal, Country, Jazz, Electronic, R&B, Indie and Folk.

The subset consists of 27,000 lyrics of songs in English distributed equally among 6 genres: country, electronic, hip-hop, jazz, pop and rock. In addition, we randomly divided the dataset into 80% training and 20% testing. 20% of the training subset was  
30 used for validation.

### 4.1.2 MoodyLyrics

For mood classification, the MoodyLyrics (ÇANO, 2017) was used. The dataset consists of 2595 music lyrics labeled according to the four quarters of Russell’s circumplex model: angry, relaxed, sad and happy.

35        The lyrics of the songs present in the dataset were labeled based on the valence and arousal value for each word present in the lyrics provided by lexicons. Through the valence and arousal value of words, music is labeled sad, happy, angry or relaxed.

### 4.1.3 Data Preprocessing

40        Song lyrics commonly contain musical annotations, such as “[CHORUS]” or “[INTRO]”, which can negatively impact the training process of the classifiers. In addition, it is necessary to standardize the data for subsequent steps. They were used as pre-processing techniques in Chapter 2 on both datasets.

- **Special Character Removal:** We remove special characters from the song lyrics using a regular expression.
- 45    • **Lower case transformation (normalization):** All the words were transformed into lowercase to uniformization and consistency.
- **Stopwords removal:** We use a stop words list provided by the NLTK python library which contains the most common English words. We remove every word present in the list from the song lyrics. We also removed the music annotations
- 50    “[INTRO]” and “[CHORUS]”.
- **Lemmatization:** We used the WordNet Lemmatizer algorithm to reduce the inflected words to their base form. In this way, the words “songs” and “song” are reduced to a unique token “song”, reducing the feature redundancy and the dictionary size.

## 55 4.2 Feature Extraction

Sequentially from data pre-processing, it is necessary to apply feature extraction techniques. It is important to note that machine learning algorithms cannot be trained directly with texts. It is necessary to provide a fixed-size entry. Therefore, it is necessary to perform a processing step to extract relevant information. This process is named feature

60    extraction.

In this work, we use two categories of feature extraction algorithms. The first category is based on representing a song based on its word count. For this, we use the

algorithms Bag-of-Words and Term Frequency - Inverse Document Frequency. The second category relies on the word semantic value.

#### 65 4.2.1 Bag of Words

The [Bag of Words \(BoW\)](#) model is a simple text representation model that is widely used in the area of [Natural Language Processing \(NLP\)](#). The model consists of representing textual information in numerical vectors, to be later used in machine learning algorithms.

70 To obtain the vector representation of lyrics in the [BoW](#) model, it is necessary to build a vocabulary of words. The word vocabulary consists of a set of words present in the song lyrics collection. In this research, we built a vocabulary consisting of 8,000 unique words. In this way, each song's lyrics will be represented as an 8000-dimensional vector. Each dimension will indicate whether the lyrics use a word from the dictionary.

75 To build the word dictionary, we ignored 5% of the most used words. The exclusion of these words suggests that very frequent values do not contain relevant information to the classification task.

In addition, we use two variations of [BoW](#): Binary and Word Count. As mentioned earlier, the binary version consists of identifying whether a word from the  
80 dictionary is present or absent in the lyrics. Non-binary [BoW](#), on the other hand, captures how many times a dictionary word appears in song lyrics.

#### 4.2.2 Term Frequency - Inverse Document Frequency

The feature extraction method [Term Frequency - Inverse Document Frequency \(TF-IDF\)](#) was also used for vector representation of songs lyrics. In comparison to [BoW](#),  
85 [TF-IDF](#) uses a weighting factor, denoting the importance of a given word in relation to the document and the corpus.

In order to compare the feature extraction methods, we parameterized the construction of the dictionary in the same way as the [BoW](#) model. We used the 8.000 most frequent words in the corpus, excluding 5% of the most common words.

90 Through the use of information extraction methods based on counting or appearances of words, it is possible to verify if a certain musical genre or mood uses specific words, which are not used in other categories.

#### 4.2.3 GloVe

In addition to the methods of extracting features based on frequency or word  
95 appearances, we use a semantic feature extractor. Semantic feature extraction methods

are capable of representing textual data in numeric numbers, which carry semantic values. For this, we use the [Global Vectors for Word Representation \(GloVe\)](#) method.

Through the use of [GloVe](#), it is possible to analyze and verify the existence of semantic patterns in the lyrics. It is possible to verify if songs from a certain category tend to address a subject. Through the analysis of the semantic value, we can verify if most of the Country lyrics tend to have words related to farm, for example.

As discussed in Section 2, the generation of word embeddings of the [GloVe](#) model is performed through the word co-occurrence matrix. However, for a large number of words, obtaining the co-occurrence matrix and sequentially the embedding vectors becomes an extremely computationally expensive process.

In this work, we use the mapping of embeddings provided by Pennington et. al<sup>1</sup>. The mapping consists of providing 50 dimension embedding vectors for 400,000 different words.

In sequence, for each lyric, we removed words that did not have a mapping for their respective vector representation. Then, we used the first 128 words and mapped them to their vector representation using [GloVe](#).

## 4.3 Music Genre and Mood Classification

Based on the features extracted from the lyrics in the previous step, we divided the data into a training set, containing 80% of the database examples and a test set, containing 20% of the examples. We use two different classifiers for the classification task: [Support Vector Machines \(SVM\)](#) and [Artificial Neural Network \(ANN\)](#).

### 4.3.1 Support Vector Machine

We used the [SVM](#) classifier for labeling data from the word frequency-based feature extraction methods: [BoW](#) and [TF-IDF](#).

In the model training phase, we utilized the Grid Search cross-validation algorithm, in order to obtain the combination of hyperparameters that maximizes the model's classification accuracy. The grid-search algorithm consists of training the model for each combination of pre-defined hyperparameters. We performed the grid search varying the  $C$  and gamma *hyperparameters*.

Finally, in the test process, we used the best combination of hyperparameters to classify the examples not yet presented to the classifier.

---

<sup>1</sup> <https://nlp.stanford.edu/projects/glove/>

### 4.3.2 Artificial Neural Networks

The training process of the ANN was performed for 20 epochs, using the Adam optimizer and the cross-entropy as a loss function. Furthermore, the training set was divided into 80% for training and 20% for validation. On the test process, the ANN classified the entries present in the test set. Finally, the accuracy and confusion matrix of the ANN prediction for the test set were computed in the classification process.

Additionally, we use two distinct ANN architectures to perform the classification task. The first architecture consists of 4 dense layers, which were used to classify features based on word frequency.

The second architecture is constituted by the embedding layer, a flatten layer and a dense layer, which was used to classify the word embedding features. For both architectures, we used the softmax activation function with the number of neurons equivalent to the number of classes present in the dataset: 6 for the music genres dataset and 4 for the mood dataset.

## 4.4 Exploratory Analysis

In addition to using SVM and ANN models to classify textual features, we performed an exploratory data analysis. Machine learning-based classifiers have great power of generalization and pattern recognition. However, the predictions made by these models are often not comprehensible to humans. We used exploratory analysis to analyze the result of the classifiers in the task of predicting musical genre and mood.

We performed two analyzes on the data. The first experiment consists of visualizing the GloVe word embedding vectors. In addition, we performed an analysis of topics covered in the lyrics of different music categories through topic analysis. The experiments will be explained in the following sections.

### 4.4.1 Word Embedding Visualization

The exploratory analysis through the visualization of embedding vectors aims to analyze and identify possible semantic patterns of song lyrics. The main objective of this experiment is to analyze and interpret whether lyrics in a particular category tend to use words with similar semantic meaning.

To perform the experiment, we used the embedding vectors from GloVe, along with its label. We map each word in the corpus to its respective embedding vector.

Since the embedding vectors have a high dimensionality for the semantic representation of each word, a dimensionality reduction method was necessary. We use the



160 [Isometric Mapping \(ISOMAP\)](#) ([TENENBAUM; SILVA; LANGFORD, 2000](#)) algorithm to reduce the dimension of the embedding vectors to 2.

After that, we elaborate a visualization of all the words in the database, highlighting the words that belong to the same taxonomy. By visualizing the semantic vectors of words from a specific taxonomy, it is possible to verify whether different  
165 taxonomies use words with similar meanings.

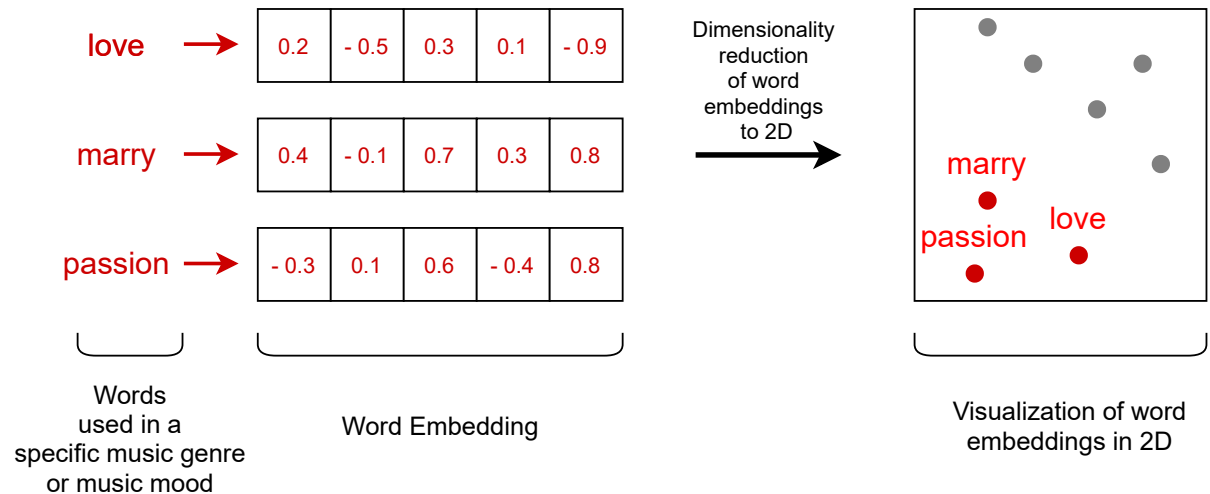


Figure 4.1 – Methodology used to visualize word embedding vectors

### 4.4.2 Topic Analysis

The second exploratory analysis experiment consists of verifying the topics covered in different musical categories through the topic analysis technique. Through this experiment, it is possible to verify if songs from specific genres tend to approach the same  
170 subject.

Initially, we extract the feature vectors using the [BoW](#) method, with a vocabulary of 8,000 words. We sequentially divide the dataset into subsets. Each subset was composed only of lyrics from a specific genre.

For the analysis and extraction of topics from each subset, we used the [Latent Dirichlet Allocation \(LDA\)](#) algorithm. Performing the model training using lyrics from  
175 subsets of each genre allows us to identify the subjects or topics covered by that music genre. Sequentially, we performed the topic extraction for each song lyrics belonging to the subset in which the [LDA](#) was trained, as illustrated in [Figure 4.2](#).

From the topics extracted from each song lyrics of a specific musical genre, it  
180 was necessary to analyze whether the lyrics of a given song tend to address the same topic. For this, we performed the calculation of the entropy of the probability distributions.

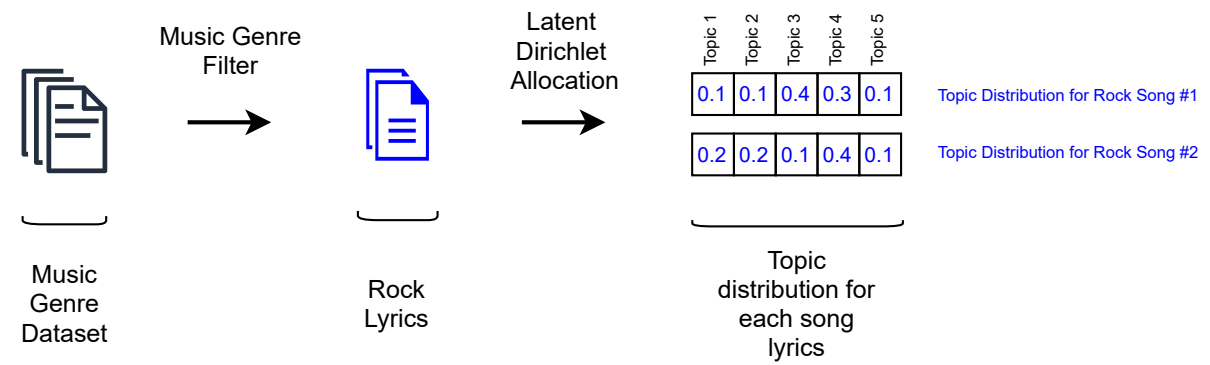


Figure 4.2 – Topic distribution (topic activation) extraction from rock lyrics.

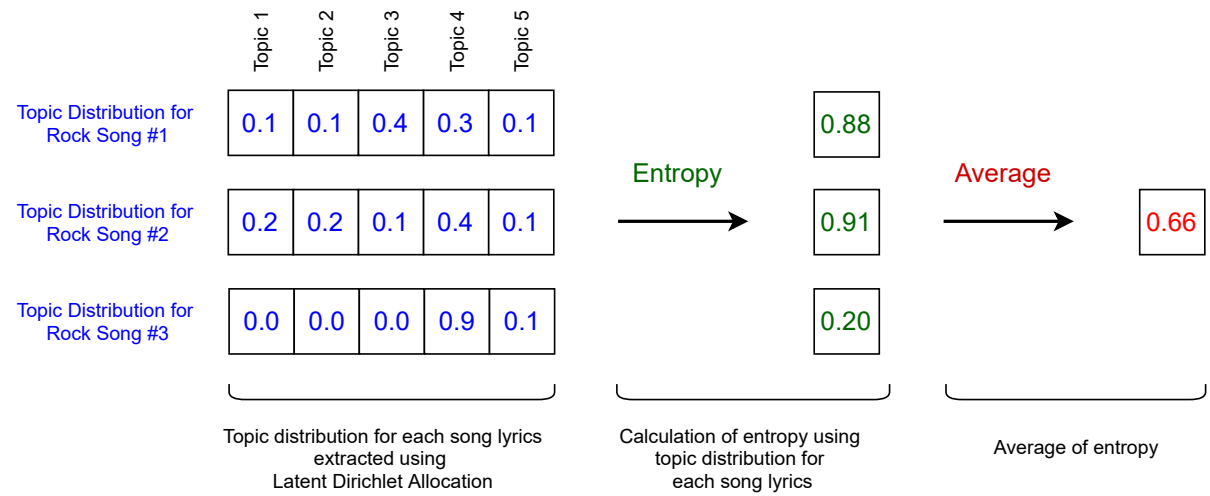


Figure 4.3 – Entropy calculated from topic activations obtained using Latent Dirichlet Allocation

A high entropy value means that song lyrics tend to vary the covered topics. On the other hand, a low entropy value indicates that song lyrics of a specific music genre tend to address the same topic.

## 5 Results and Discussion

In this section, we present the results obtained from the experiments described in Section 4. We will present the results referring to the exploratory analyses, which were performed especially to understand and explore the datasets. Furthermore, results of the gender and mood classification experiments using the machine learning algorithms [Support Vector Machines \(SVM\)](#) and [Artificial Neural Network \(ANN\)](#) together with the textual data are also presented in Section 5.2.

### 5.1 Exploratory Analysis

Exploratory analysis was used to visualize, explore and verify the existence of patterns of different musical categories in the information from song lyrics through semantic analysis and topic analysis.

#### 5.1.1 Embedding Visualization

The first exploratory analysis experiment performed was the visualization of embeddings. As discussed earlier, the main idea of this experiment is to identify the possible semantic patterns used between lyrics of different categories. That is, check whether music categories use words with similar semantic meaning.

Figure 5.1 illustrates the word semantic value used in song lyrics of the 4 genres of 6 genres present in the dataset. Each word is represented as a dot on the graph. The words were mapped to coordinates by reducing the dimensionality of the [Global Vectors for Word Representation \(GloVe\)](#) vector representation, using the [Isometric Mapping \(ISOMAP\)](#) algorithm. Blue dots represents the presence of the word in the song lyric of a specific genre and the absence is represented by a gray dot.

Through the visualization of embedding vectors, we can observe that lyrics from different musical genres lack a semantic particularity. Different music genres tend to use words with different meanings.

Since it is not possible to notice a distinction in a typical pattern of semantic values used in song lyrics, the problem of classifying music genres requires another approach besides [GloVe](#) word embedding.

Figure 5.2 illustrate the semantic values of words used in lyrics from 2 of 4 different moods present in MoodyLyrics. The mood sad contains a group of words which songs of the other moods do not use, helping to distinguish it from other classes. However,

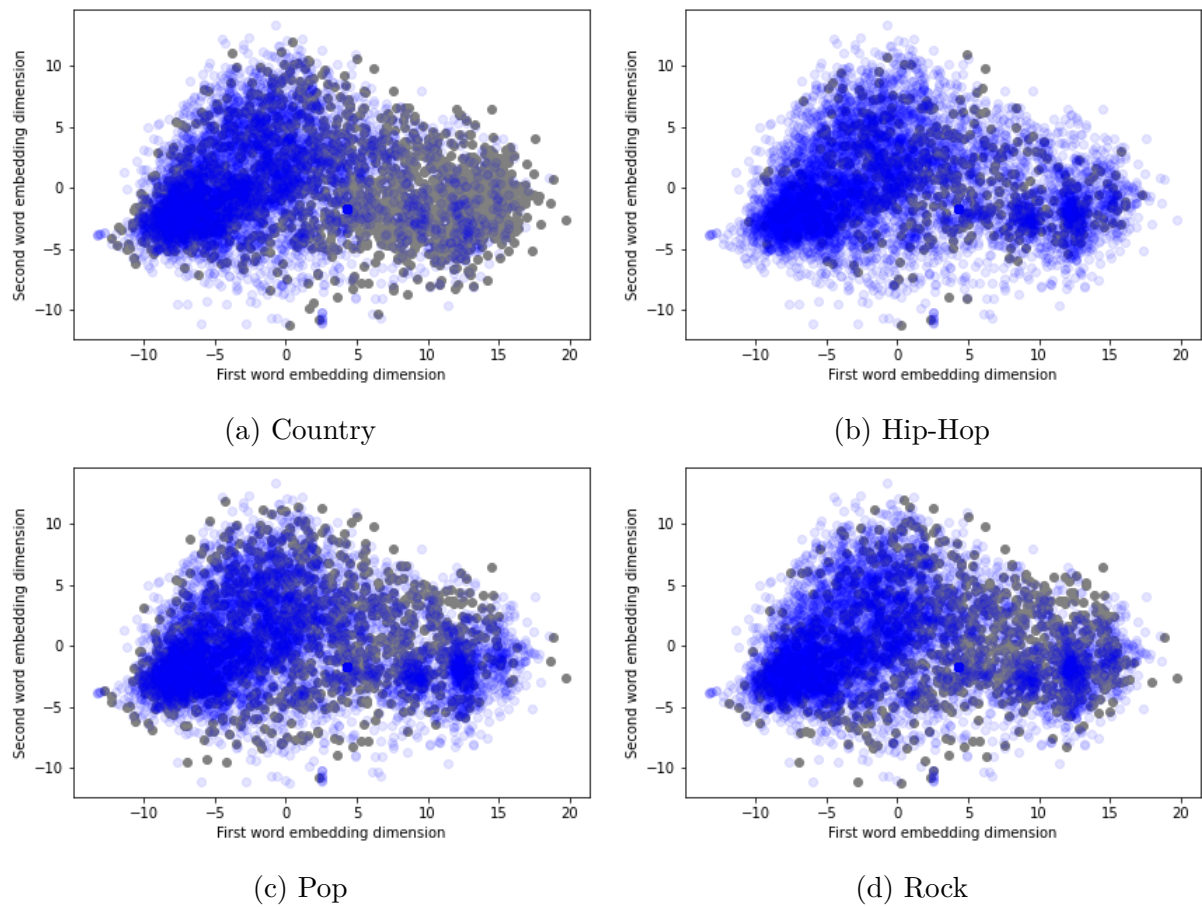


Figure 5.1 – Visualization of words used in each genre in the word embedding space. We can see that most genres use most words in the embedding space.

a group of words is present in song lyrics of the four moods. Like the classification of music genres, in the mood classification it is not possible to notice the use of typical words across all the moods, except for sad songs.

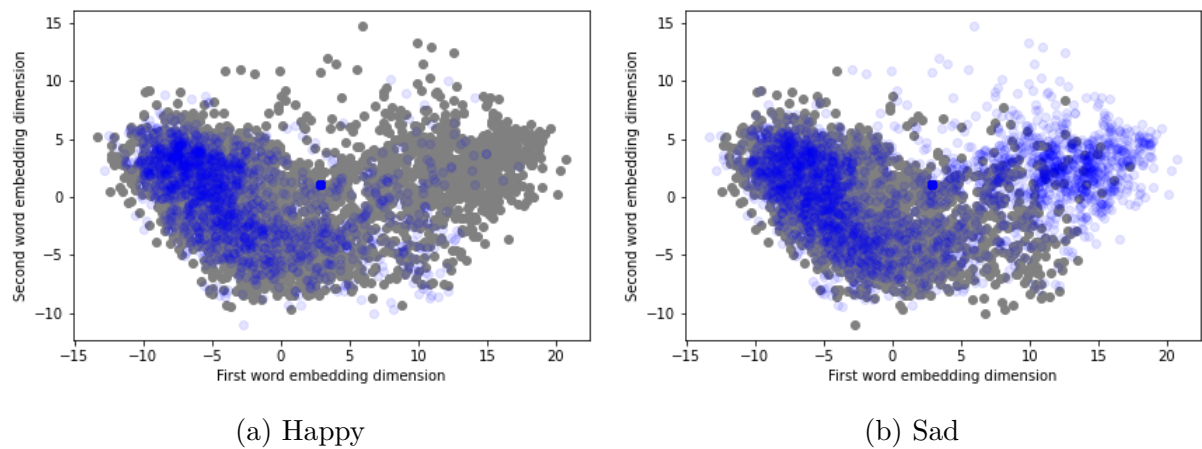


Figure 5.2 – Visualization of words used in each mood in the word embedding space

### 5.1.2 Topic Analysis

The second exploratory analysis experiment performed to analyze the dataset was topic analysis. The idea of this experiment is to verify if different music genres address specific topics in their lyrics. For this, we use the Latent Dirichlet Allocation algorithm to obtain the topics for all the music genres present in the dataset.

Figure 5.3 illustrates the average entropy of the extracted topics activations for each musical genre. Importantly, a high entropy value means that topic activations for a music genre vary widely. This means that lyrics of a particular genre tend to cover different topics or subjects.

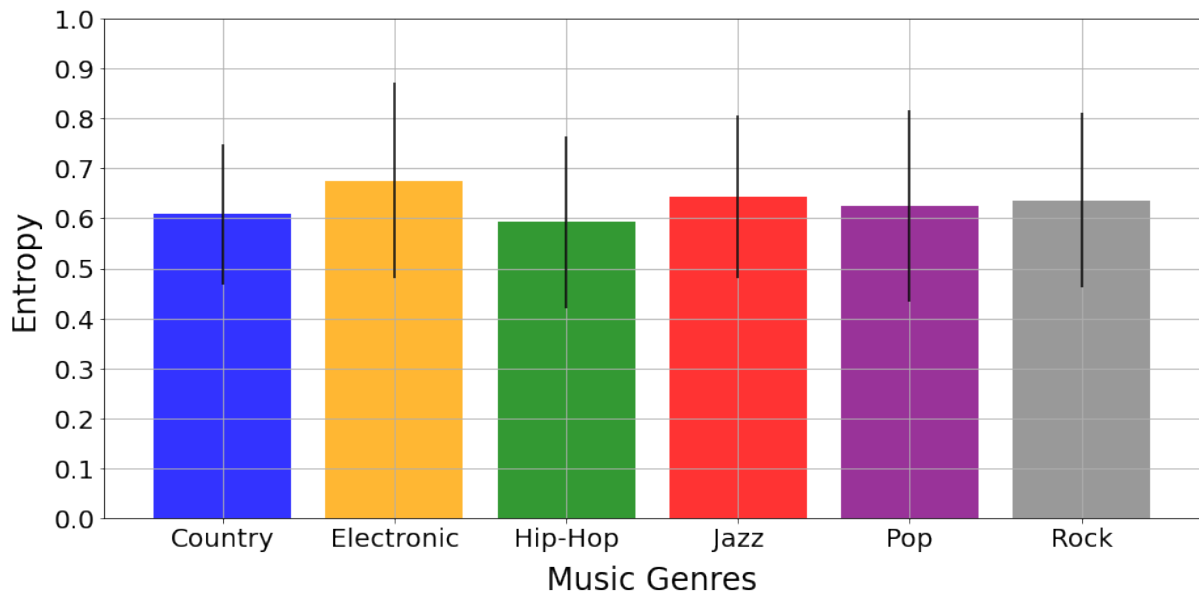


Figure 5.3 – Entropy calculated from each genre subset.

Through this experiment, we can conclude that the lyrics of a particular musical genre are not restricted in addressing a single subject or topic. We illustrate that song lyrics of the same genre address different issues through the high value of entropy.

## 5.2 Music genre and mood classification

To perform the experiment of classifying the lyrics in musical genres and moods, we used three text representation algorithms, two based on frequency or word count: [Bag of Words \(BoW\)](#) and [Term Frequency - Inverse Document Frequency \(TF-IDF\)](#) and an algorithm based on semantic value from the words: [GloVe](#). We use the extracted features in two classifiers: [SVM](#) and [ANN](#). In this section, we illustrate the results obtained from the two classifiers used.

Table 5.1 and Table 5.2 illustrate the results obtained from the classification of music genres using the MetroLyrics dataset and the mood classification using the

MoodyLyrics dataset, respectively.

Features / Model	SVM	Neural Network
BoW Binary	0.57	0.43
BoW	0.35	0.44
TF-IDF	<b>0.57</b>	0.43
Word Embedding	*	0.32

Table 5.1 – Model accuracy and text features used in genre classification experiments

Features / Model	SVM	Neural Network
BoW Binary	0.32	0.66
BoW	0.24	<b>0.74</b>
TF-IDF	<b>0.74</b>	0.72
Word Embedding	*	0.58

Table 5.2 – Model accuracy and text features used in mood classification experiments

Although the textual representation **TF-IDF** is simple and does not capture the semantic value of words, the features had the best performance using Support Vector Machine. For the task of classifying music genres, the accuracy was 0.57. For the mood  
 60 classification task, the accuracy was 0.74.

The confusion matrix for the **SVM** classifier with **TF-IDF** features, which achieves an accuracy of 0.57 are shown in Figure 5.4. Figure 5.5 illustrates the confusion matrix obtained by the **ANN** classification result utilizing word embedding. In our experiments, we noted that only Hip-Hop song lyrics uses specific words that are not present in  
 65 song lyrics of other genres. The occurrence of these typical words can contribute to the greater ease of distinguishing Hip-Hop songs compared with songs of different genres.

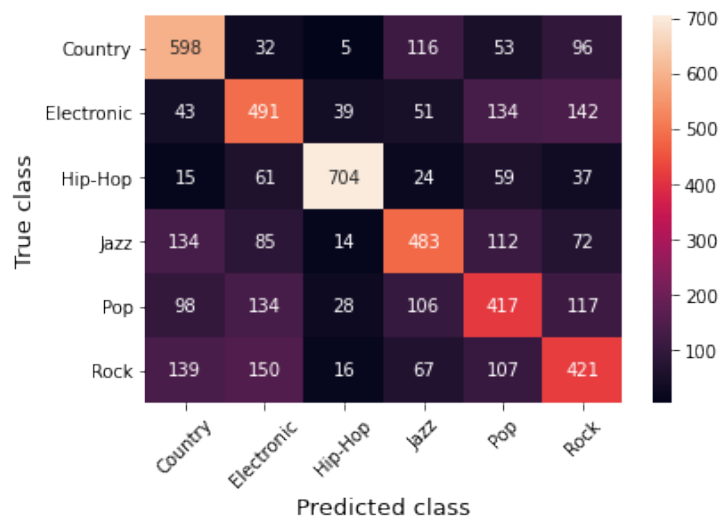


Figure 5.4 – Confusion Matrix for genre classification using SVM with TF-IDF features.

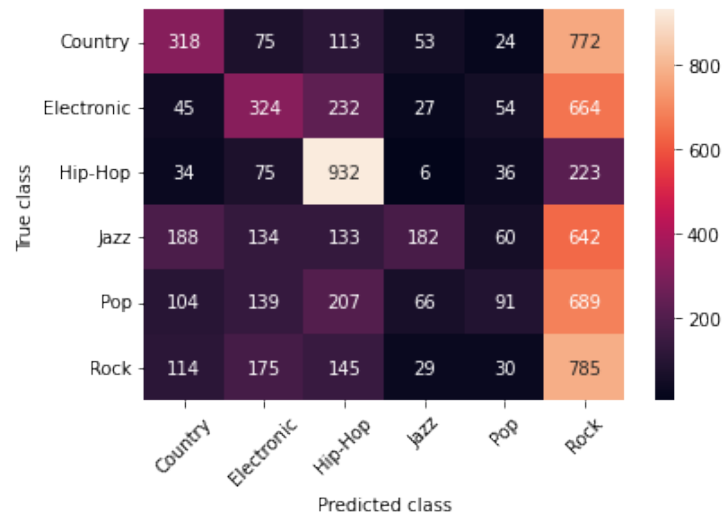


Figure 5.5 – Confusion Matrix for genre classification using Neural Network with word embedding features.

For the mood classification, ANN with word embeddings reached an accuracy of 0.58. In contrast, using the SVM classifier with TF-IDF features, we reached an accuracy of 0.74. Figure 5.6 show the confusion matrix obtained using SVM with TF-IDF features and Figure 5.7 illustrates the confusion matrix for the neural network classification result using word embedding.

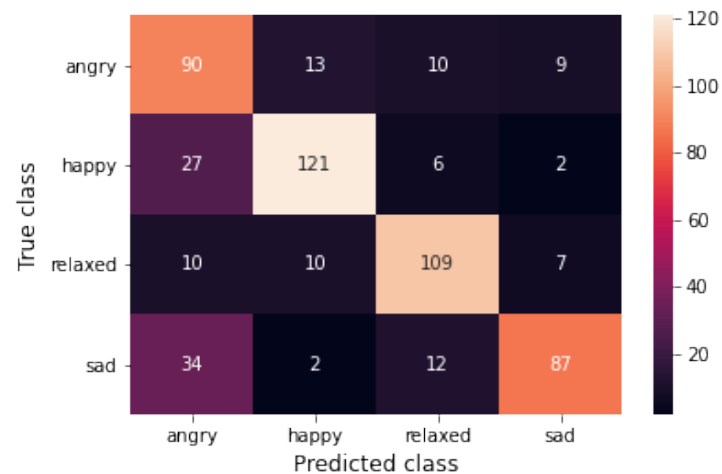


Figure 5.6 – Confusion Matrix for mood classification using SVM with TF-IDF features.

The results show that TF-IDF features have better performance compared to word embedding. This suggests that there are some specific words that have a higher frequency of use in a certain class compared to others. Therefore, the most frequent words in each class are given greater weight in the TF-IDF features, benefiting the classification task.



Figure 5.7 – Confusion Matrix for mood classification using Neural Network with word embedding features.



## 6 Conclusion and Future Work

The task of music genre classification and mood classification is a challenging task. One of the great difficulties of this task is due to the subjectivity of these categories. There is no formal definition conceptualizing which characteristics or rules a song must  
 5 have to be categorized as a certain musical genre or mood.

In this work, we explore the problems of classifying music into music genres or mood. To approach this problem, we use machine learning algorithms used to classify textual data from song lyrics.

We use the [Bag of Words \(BoW\)](#) and [Term Frequency - Inverse Document  
 10 Frequency \(TF-IDF\)](#) models to extract textual features based on word counts. In addition, we employ the [Global Vectors for Word Representation \(GloVe\)](#) word embedding model to extract the semantic value of the words.

Lyric classification resulted in an accuracy of 57% in categorizing six distinct music genres: Country, Electronic, Hip-Hop, Jazz, Pop and Rock. We got 74% accuracy  
 15 for the mood classification in classifying angry, happy, relaxed and sad moods. The results indicate that the characteristics used from song lyrics based on word frequency or semantic values are not efficient descriptive for this task.

We also performed an exploratory analysis of the dataset. We performed the exploratory analysis to identify possible patterns of lyrics of the same category. Furthermore,  
 20 the exploration of the data allowed us to elucidate the performance of the classifiers.

For this, we used the visualization of semantic vectors provided by [GloVe](#) and the analysis of topics covered in different genres through [Latent Dirichlet Allocation \(LDA\)](#). The experiments indicate that distinct music genres or moods lack a particularity. Lyrics of the same music genre or mood are not restricted to addressing only one subject or using  
 25 words with the same meaning.

We can conclude that song lyrics are not features discriminative enough to distinguish music genre or mood. This is because there is a significant overlap between the words used in songs from different genres or moods. Henceforth, although it could be possible to emerge models that are able to distinguish among these categories, our  
 30 evidence supports the idea that high accuracies, in these problems, could be more due to specific characteristics of datasets than to model capabilities.

## 6.1 Future Work

In this research, we use two distinct approaches to extract feature for data analysis and classification. As mentioned before, we use the [BoW](#), [TF-IDF](#) and [GloVe](#) feature extractors. Future works exploring textual feature extraction methods to elucidate the lack of relationship between song lyrics and musical genres or moods can be performed.

Performing exploratory analysis and lyrics classification using different machine learning algorithms would be interesting. In our experiments, we performed the visualization of the embedding vectors by reducing dimensionality using the [Isometric Mapping \(ISOMAP\)](#) algorithm. The visualization obtained by [ISOMAP](#) does not show any particularities of genres or moods. Perhaps, another dimensionality reduction algorithm can be used to analyze the data and verify if it is possible to identify particularities of each music genre or mood.

# Bibliography

ABIODUN, O. I. et al. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, Elsevier, v. 4, n. 11, p. e00938, 2018. Cited on page [24](#).

ALPAYDIN, E. *Introduction to machine learning*. Massachusetts: MIT press, 2020. Cited on page [19](#).

ANGLUIN, D.; SMITH, C. H. Inductive inference: Theory and methods. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 15, n. 3, p. 237–269, 1983. Cited on page [19](#).

BISHOP, C. M. *Pattern recognition and machine learning*. Manhattan: Springer, 2006. Cited 2 times on pages [19](#) and [20](#).

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *the Journal of machine Learning research*, JMLR. org, v. 3, p. 993–1022, 2003. Cited on page [32](#).

CANICATTI, A. Song genre classification via lyric text mining. In: THE STEERING COMMITTEE OF THE WORLD CONGRESS IN COMPUTER SCIENCE. *Proceedings of the International Conference on Data Science (ICDATA)*. Las Vegas, Nevada, USA: DMIN, 2016. p. 44. Cited on page [18](#).

COVER, T.; HART, P. Nearest neighbor pattern classification. *IEEE transactions on information theory*, IEEE, v. 13, n. 1, p. 21–27, 1967. Cited on page [34](#).

CUNNINGHAM, J. P.; GHAHRAMANI, Z. Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research*, JMLR. org, v. 16, n. 1, p. 2859–2900, 2015. Cited on page [33](#).

DONOHU, D. L. et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, Citeseer, v. 1, n. 2000, p. 32, 2000. Cited on page [33](#).

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification*. 2. ed. New York: Wiley, 2001. ISBN 978-0-471-05669-0. Cited on page [20](#).

DUMAIS, S. T. Latent semantic analysis. *Annual review of information science and technology*, Wiley Online Library, v. 38, n. 1, p. 188–230, 2004. Cited on page [32](#).

EKMAN, P. Expression and the nature of emotion. *Approaches to emotion*, v. 3, n. 19, p. 344, 1984. Cited on page [15](#).

FALEIROS, T.; LOPES, A. *Modelos probabilístico de tópicos: desvendando o Latent Dirichlet Allocation*. 2016. Cited on page [33](#).

FAUSETT, L. *Fundamentals of neural networks: architectures, algorithms, and applications*. [S.l.]: Prentice-Hall, Inc., 1994. Cited on page [24](#).

GEMAN, S.; GEMAN, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, n. 6, p. 721–741, 1984. Cited on page [33](#).

- HARRIS, Z. S. Distributional structure. *Word*, Taylor & Francis, v. 10, n. 2-3, p. 146–162, 1954. Cited on page 28.
- HAYKIN, S.; NETWORK, N. A comprehensive foundation. *Neural networks*, v. 2, n. 2004, p. 41, 2004. Cited on page 20.
- HOFMANN, T. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*, 2013. Cited on page 32.
- HU, X.; DOWNIE, J. S. When lyrics outperform audio for music mood classification: A feature analysis. In: . Utrecht, Netherlands: ISMIR, 2010. p. 619–624. Cited on page 15.
- JOLLIFFE, I. Principal component analysis. *Encyclopedia of statistics in behavioral science*, Wiley Online Library, 2005. Cited on page 34.
- KRÖSE, B. et al. An introduction to neural networks. Citeseer, 1993. Cited on page 25.
- KUMAR, A.; RAJPAL, A.; RATHORE, D. Genre classification using word embeddings and deep learning. In: IEEE. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. Bangalore, India, 2018. p. 2142–2146. Cited on page 15.
- LALWANI, T. et al. Implementation of a chatbot system using ai and nlp. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Volume-6, Issue-3*, 2018. Cited on page 27.
- LI, C.-G. et al. A version of isomap with explicit mapping. In: . Dalian, China: IEEE, 2006. p. 3201 – 3206. Cited on page 34.
- LI, S.; GUO, G. Content-based audio classification and retrieval using svm learning. 01 2000. Cited on page 13.
- LIU, H.; MOTODA, H. *Feature selection for knowledge discovery and data mining*. [S.l.]: Springer Science & Business Media, 2012. v. 454. Cited on page 33.
- LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007. Cited 2 times on pages 9 and 24.
- MAATEN, L. V. D. et al. Dimensionality reduction: a comparative. *J Mach Learn Res*, v. 10, n. 66-71, p. 13, 2009. Cited on page 33.
- MAYBURY, M. *Advances in automatic text summarization*. Massachusetts: MIT press, 1999. Cited on page 27.
- MAYER, R.; NEUMAYER, R.; RAUBER, A. Combination of audio and lyrics features for genre classification in digital audio collections. In: . Vancouver, British Columbia: ACM, 2008. p. 159–168. Cited on page 15.
- MAYER, R.; RAUBER, A. Music genre classification by ensembles of audio and lyrics features. In: . Miami, Florida: University of Miami, 2011. p. 675–680. Cited on page 14.
- MCKINNEY, M.; BREEBAART, J.; (WY, P. Features for audio and music classification. 11 2003. Cited on page 13.

MIDDI, V. S. R.; RAJU, M.; HARRIS, T. A. Machine translation using natural language processing. *MATEC Web of Conferences*, v. 277, p. 02004, 01 2019. Cited on page 27.

NAKARIYAKUL, S.; CASASENT, D. An improvement on floating search algorithms for feature subset selection. *Pattern Recognition*, v. 42, p. 1932–1940, 09 2009. Cited on page 34.

NASREEN, S. A survey of feature selection and feature extraction techniques in machine learning. In: . London, UK: IEEE, 2014. Cited on page 33.

OGIHARA, M. et al. The semantic shapes of popular music lyrics: Graph-based representation, analysis, and interpretation of popular music lyrics in semantic natural language embedding space. In: IEEE. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Orlando, Florida, 2018. p. 1249–1254. Cited on page 29.

PACHET, F.; CAZALY, D. et al. A taxonomy of musical genres. In: CITESEER. *RIAO*. Paris, FRA: LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2000. p. 1238–1245. Cited on page 16.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543. Cited 3 times on pages 9, 30, and 31.

RAMÍREZ, J.; FLORES, M. J. Machine learning for music genre: multifaceted review and experimentation with audioset. *Journal of Intelligent Information Systems*, Springer Science and Business Media LLC, v. 55, n. 3, p. 469–499, Nov 2019. ISSN 1573-7675. Cited on page 14.

Ren, J.; Wu, M.; Jang, J. R. Automatic music mood classification based on timbre and modulation features. *IEEE Transactions on Affective Computing*, v. 6, n. 3, p. 236–246, 2015. Cited on page 15.

RODRÍGUEZ, I. S. Text similarity by using glove word vector representations. Universitat Politècnica de València, 2017. Cited on page 31.

RUSSELL, J. A. A circumplex model of affect. *Journal of personality and social psychology*, American Psychological Association, v. 39, n. 6, p. 1161, 1980. Cited 2 times on pages 8 and 16.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, v. 24, n. 5, p. 513–523, 1988. ISSN 0306-4573. Cited on page 28.

SCARINGELLA, N.; ZOIA, G.; MLYNEK, D. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, IEEE, v. 23, n. 2, p. 133–141, 2006. Cited on page 16.

SCHENK, J.; KAISER, M.; RIGOLL, G. Selecting features in on-line handwritten whiteboard note recognition: Sfs or sffs? In: *2009 10th International Conference on Document Analysis and Recognition*. Barcelona, Spain: IEEE. Cited on page 34.

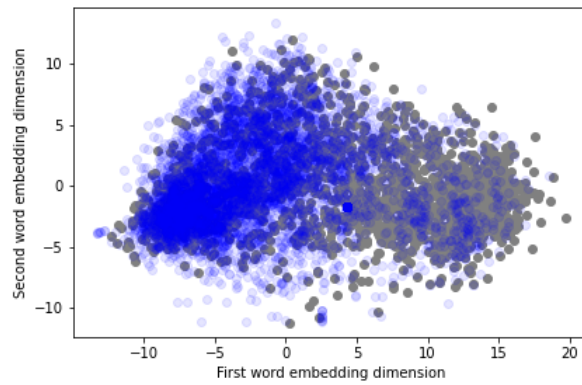
- Sharma, V. et al. Sentiments mining and classification of music lyrics using sentiwordnet. In: *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. Indore, India: IEEE, 2016. p. 1–6. Cited on page [15](#).
- SIRIKET, K.; SA-ING, V.; KHONTHAPAGDEE, S. Mood classification from song lyric using machine learning. In: *2021 9th International Electrical Engineering Congress (iEECON)*. Pattaya, Thailand: IEEE, 2021. p. 476–478. Cited on page [18](#).
- TENENBAUM, J. B.; SILVA, V. D.; LANGFORD, J. C. A global geometric framework for nonlinear dimensionality reduction. *science*, American Association for the Advancement of Science, v. 290, n. 5500, p. 2319–2323, 2000. Cited 2 times on pages [34](#) and [41](#).
- TOMAN, M.; TESAR, R.; JEZEK, K. Influence of word normalization on text classification. *Proceedings of InSciT*, v. 4, p. 354–358, 2006. Cited on page [28](#).
- UYVAL, A. K.; GUNAL, S. The impact of preprocessing on text classification. *Information processing & management*, Elsevier, v. 50, n. 1, p. 104–112, 2014. Cited on page [27](#).
- WANG, B. et al. Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, Cambridge University Press, v. 8, p. e19, 2019. Cited 2 times on pages [29](#) and [30](#).
- WEINER, B.; GRAHAM, S. An attributional approach to emotional development. *Emotions, Cognition, and Behavior*, p. 167–191, 01 1984. Cited on page [15](#).
- YING, T. C.; DORAISAMY, S.; ABDULLAH, L. N. Genre and mood classification using lyric features. In: *2012 International Conference on Information Retrieval & Knowledge Management*. Kuala Lumpur, Malaysia: IEEE, 2012. p. 260–263. Cited 2 times on pages [15](#) and [18](#).
- ZAAANEN, M. V.; KANTERS, P. Automatic mood classification using tf\* idf based on lyrics. In: . Utrecht, Netherlands: ISMIR, 2010. p. 75–80. Cited on page [15](#).
- ÇANO, E. Moodylyrics: A sentiment annotated lyrics dataset. In: . Hong Kong, China: ISMSI, 2017. p. 118–124. Cited on page [37](#).

## Annex

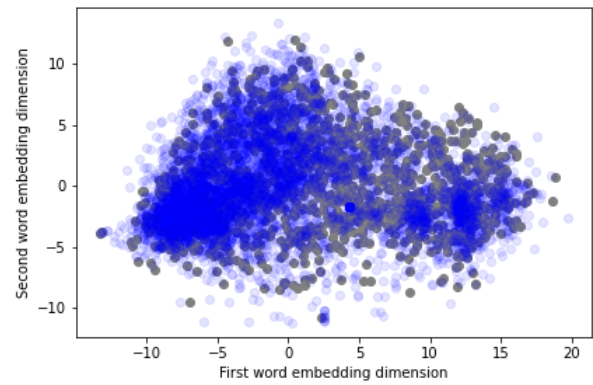


# ANNEX A – Music Genre Classification

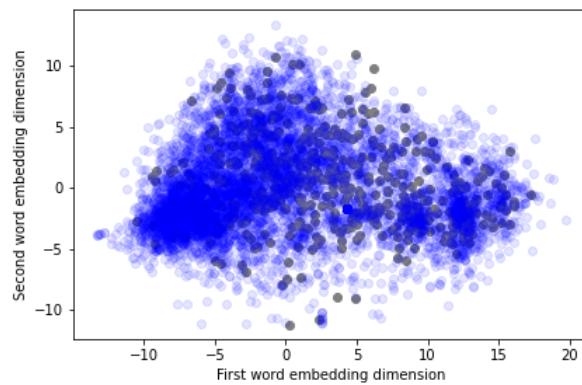
## Word Embedding Visualization



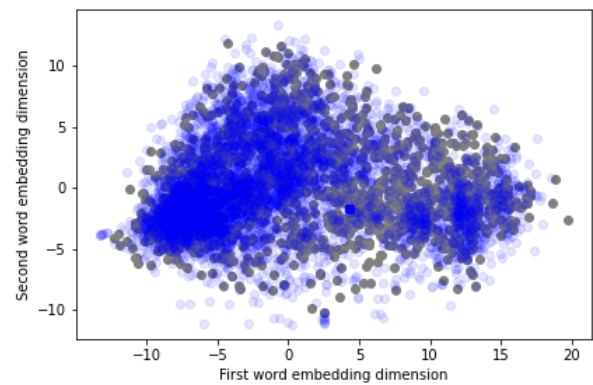
(a) Country



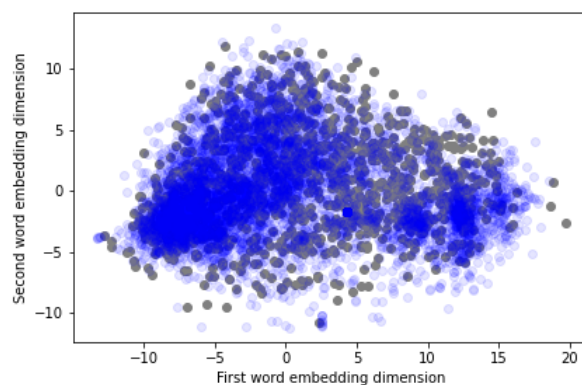
(b) Electronic



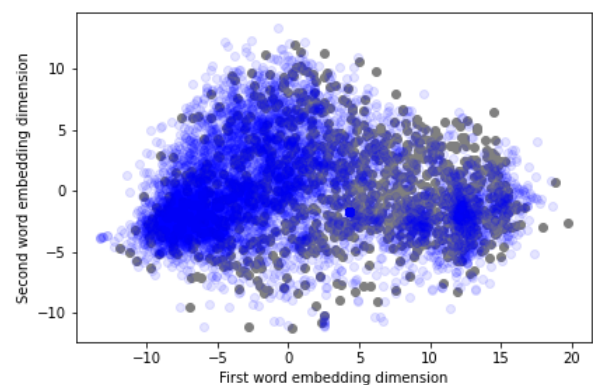
(c) Hip-Hop



(d) Jazz



(e) Pop



(f) Rock

Figure A.1 – Visualization of words used in each genre in the word embedding space. We can see that most genres use most words in the embedding space.



# ANNEX B – Music Mood Classification

## Word Embedding Visualization

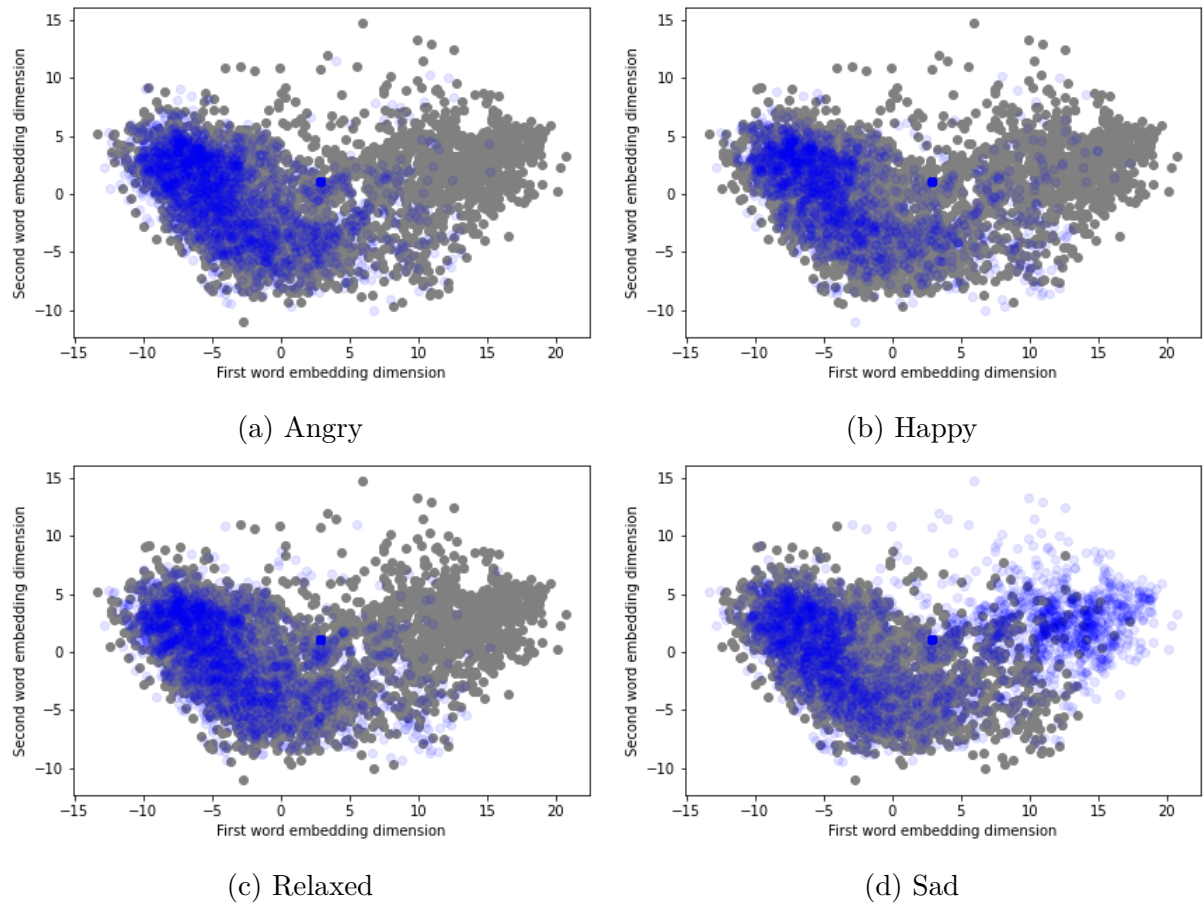


Figure B.1 – Visualization of words used in each mood in the word embedding space