MUSIC EMOTION RECOGNITION ALGORITHM (MERA) USING DEEP LEARNING

BONFACE MARTIN OYWA N11/3/0486/017

Software project in partial fulfilment of the requirements for award on Bachelor of Science in Computer Science degree of Laikipia University

OCTOBER 2021

# DECLARATIONS

I hereby declare that this project is my work and has nor been submitted to any other university for purpose of examination. All the information given is my own and all the cited sources are quoted and acknowledged accordingly.

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# RECOMMENDATION

The project “Music Emotion Recognition Algorithm (MERA) for Music Recommender Systems using Deep Learning” has been present to the Computing and Informatics Department of Laikipia University. We have received the thesis and recommend it to be accepted in partial fulfilment for the Bachelor of Science in Computer Science.

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Madam Lorna Ogake

Department of Computing and Informatics, Laikipia University

# COPYRIGHT

All rights reserved no part of this thesis may be reproduced or stored in any retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying and recording without prior written permission of the author or Laikipia University on that behalf.

# ACKNOWLEDGMENT

# DEDICATION

I dedicate this project to the Department of Computing and Informatics, Laikipia University. To my supervisor Madam Lorna Ogake who patiently listened to mu presentations and gave me the guidance I needed to complete the project and make it a success.

To my fellow students who supported me through words of encouragement and technical assistance whenever I reached out to them. I cannot also forget my mother and grandparents who contributed immensely to my learning process. The financial and emotional aid ensured that I had peace of mind as I worked on my project and had access to all the necessary materials.

# ABSTRACT

Music Emotion Recognition is a concept in Music Information Retrieval which hasn’t been around for a very long time, that is involved in determining or classifying emotions or mood that humans perceive in music using computer programs.

In this project I focused on the use of Deep Learning, a field in Artificial Intelligence that applies a concept called Deep Neural Networks (DNNs) for learning representations or patterns from data, which helps the networks performs their specified tasks such as classification of labels. Originally DNNs were only used in classification of images, but can now be used even on classification of text as part of Natural Language Processing (NLP), more specifically Natural Language Understanding. Both tasks will be used for emotion or mood recognition, later described in this project.

The project is further built on the concepts of Valence and Arousal in music, which represent negative to positive moods, and calm to energetic moods respectively. Both the audio and lyrics of a music are used since Arousal correlates well to audio while Valence correlates well to both audio and lyrics. The goal here is to use Deep Learning for classification of mel-spectrograms defined from music audio in image classification, and classification of music lyrics in text classification then averaging out the results to come up with a prediction of emotion. Two models will therefore be used in this project.

TABLE OF CONTENTS

[DECLARATIONS I](#__RefHeading___Toc1629_2478191445)

[RECOMMENDATION II](#__RefHeading___Toc1631_2478191445)

[COPYRIGHT III](#__RefHeading___Toc1633_2478191445)

[ACKNOWLEDGMENT IV](#__RefHeading___Toc1635_2478191445)

[DEDICATION V](#__RefHeading___Toc1637_2478191445)

[ABSTRACT VI](#__RefHeading___Toc1639_2478191445)

[CHAPTER 1 – INTRODUCTION 1](#__RefHeading___Toc1641_2478191445)

[1.1 BACKGROUND OF STUDY 1](#__RefHeading___Toc1643_2478191445)

[1.2 PROBLEM DEFINITION 1](#__RefHeading___Toc1645_2478191445)

[1.3 SYSTEM OBJECTIVES 1](#__RefHeading___Toc1647_2478191445)

[1.3.1 GENERAL OBJECTIVE 1](#__RefHeading___Toc1649_2478191445)

[1.3.2 SPECIFIC OBJECTIVES 2](#__RefHeading___Toc1651_2478191445)

[1.4 JUSTIFICATION 2](#__RefHeading___Toc1653_2478191445)

[1.5 PROJECT SCOPE 2](#__RefHeading___Toc1655_2478191445)

[1.6 RESEARCH LIMITATIONS 2](#__RefHeading___Toc1657_2478191445)

[CHAPTER 2 – LITERATURE REVIEW 3](#__RefHeading___Toc1659_2478191445)

[2.1 INTRODUCTION 3](#__RefHeading___Toc1661_2478191445)

[2.2 MUSIC EMOTION RECOGNITION (MER) 3](#__RefHeading___Toc1663_2478191445)

[2.3 DEEP LEARNING 4](#__RefHeading___Toc1665_2478191445)

[2.4 TRANSFER LEARNING 4](#__RefHeading___Toc1667_2478191445)

[2.5 BIDIRECTIONAL ENCODER REPRESENTAIONS TRANSFORMER (BERT) 4](#__RefHeading___Toc1669_2478191445)

[CHAPTER 3 – METHODOLOGY 5](#__RefHeading___Toc1671_2478191445)

[3.1 DATA COLLECTION AND ANALYSIS 5](#__RefHeading___Toc1673_2478191445)

[3.1.1 COLLECTION OF MUSIC LYRICS 6](#__RefHeading___Toc1675_2478191445)

[3.1.2 COLLECTION OF SONG AUDIO 7](#__RefHeading___Toc1677_2478191445)

[3.1.3 QUADRANTS AS LABELS 8](#__RefHeading___Toc1679_2478191445)

[3.1.4 CLEANING DATA 8](#__RefHeading___Toc1681_2478191445)

[3.2 MODEL ARCHITECTURES, TRAINING AND EVALUATION 9](#__RefHeading___Toc1687_2478191445)

[3.2.1 AUDIO MODEL 9](#__RefHeading___Toc1689_2478191445)

[3.2.2 LYRICS MODEL 10](#__RefHeading___Toc1691_2478191445)

[3.3 SOFTWARE REQUIREMENTS 11](#__RefHeading___Toc1705_2478191445)

[3.4 HARDWARE REQUIREMENTS 11](#__RefHeading___Toc1685_2478191445)

[CHAPTER 4 – PROJECT PRESENTATION 12](#__RefHeading___Toc1695_2478191445)

[4.1 DATA FLOW DIAGRAMS (DFDs) 12](#__RefHeading___Toc1697_2478191445)

[4.2 PROJECT SCREENSHOTS 14](#__RefHeading___Toc1699_2478191445)

[CHAPTER 5 – RECOMMENDATIONS AND CONCLUSION 15](#__RefHeading___Toc1701_2478191445)

# CHAPTER 1 – INTRODUCTION

This chapter gives an introduction to music recommendation. The chapter further gives a background information of the project and a problem statement, as well as the objectives and a justification of the project if implemented.

# 1.1 BACKGROUND OF STUDY

Music recommendation has come a long way such that a single search of a song almost guarantees you continuous stream of other songs that you might like, and almost always do. These recommender systems usually use techniques such as collaborative filtering, and audio models to recommend music. This means that the music that gets recommended always has similarities with music you already listen to or what group of users (in collaborative filtering) you belong to.

Emotions in humans usually have a tendency to be erratic, and recommender systems sometimes won’t always be able to pick up on this since a user;s current emotional state may force them to deviate from the usual user space or profile the system has built on them, meaning that a song being recommended may not be able to fit in well with a user’s current emotion, thus using music emotion recognition using seems like something that will always be able to tailor music based on emotions, and be used by existing recommender systems to determine which songs based on emotion should it recommend.

# 1.2 PROBLEM DEFINITION

For most music streaming companies, the above mentioned methods of music recommendation are mainly for financial gain, so they tend to be very effective in terms of getting music in front of the users of their platform, thus there is some margin of error in terms of experience they are willing to forgo regardless of how it affects a users experience.

From my experience, their systems usually lack the ability to create recommendations that necessarily have the emotional impact which as a listener, I may be in search of, especially when I need to slip into and out of certain moods.

# 1.3 SYSTEM OBJECTIVES

## 1.3.1 GENERAL OBJECTIVE

To study the techniques of Deep Learning applications in music emotion recognition and how they can be used by recommender systems.

## 1.3.2 SPECIFIC OBJECTIVES

1. To implement a music emotion recognition algorithm using audio mel-spectrograms and lyrics from songs.
2. To apply the algorithm in the creation playlists or clusters based on emotion.
3. To create two models whose outputs can be used to classify new music’s cluster.
4. To explain how existing recommender systems can tap into these models for better music recommendation based on emotions or moods.

# 1.4 JUSTIFICATION

Most users have a tendency of searching for music that is in sync with their current mood or emotional state, and since music streaming platforms usually recommends music based on past listens and a listener’s similarity with others, they usually miss to meet these states such that a listener gets presented with a mix-match of what they might like which may deviate from their current interests emotionally.

This project’s justification is tied in well into using Deep Learning to aid in music classification or clustering and creation of playlists or clusters that the streaming platforms can tap into to recommend music based on emotion.

# 1.5 PROJECT SCOPE

This project only focuses on the creating of the models, their deployment on a web application and their use in clustering music based on their predicted value. No recommeder system would be built but as per specific objective iv., an explanation on how they can be used would be discussed.

# 1.6 RESEARCH LIMITATIONS

The main challenges encountered during research or study included the following;

1. Getting clear access to how their systems currently handle recommendation of music based on emotions since this may be considered confidential as it’s may be a revenue generating feature.
2. Accessing music audio and lyrics was limited due to copyright issues, so data collected may have not been sufficient enough to get very accurate models.
3. Most lyrics collected were only 30% of the actual song lyrics, thus mapping 30% of a song to actual audio was challenging thus entire audio was used.

# CHAPTER 2 – LITERATURE REVIEW

# 2.1 INTRODUCTION

This section presents a comprehensive overview of Music Information Retrieval (MIR) and its ties to Music Emotion Recognition (MER), Deep Learning (DL), and an in-depth look into Transfer Learning, and Bidirectional Encoder Representations Transformer (BERT) for multi-class text classification in Natural Language Processing, which are heavily applied in this project.

# 2.2 MUSIC EMOTION RECOGNITION (MER)

Music Information Retrieval (MIR) is an interdisciplinary science of retrieving information from music, whose aim is to develop computational tools for processing, searching, organizing, and accessing music-related data.

The field of Music Emotion Recognition (MER) has been evaluated consistently since 2007 in the Music Information Retrieval Evaluation Exchange (MIREX) Audio Mood Classification task. This task consists of the classification of audio into five mood categories or clusters, containing different emotional tags. Emotions in this context refers to what listeners perceive since; 1. It’s relatively less influenced by situational factors of listening e.g environment, and 2. Listeners are generally consistent in their ratings of the emotional expression of music (1909.05882).

In terms of definition of emotions, the following attributes and characteristics of emotions were considered; 1. Emotions appear to vary their intensity e.g irritation – rage, 2. Emotions appear to involve distinct qualitative feelings e.g feeling of being afraid – nostalgic, 3. Some pairs of emotions appear more similar than others e.g joy and contentment versus joy and disgust, and 4. Some emotions appear to have opposites e.g happiness – sadness, love – hate, calm – worry. These attributes altogether led to the definition of the following two approaches for representing emotions;

1. Categorical approach where emotions are represented as categories distinct from each other, i.e happiness, sadness, anger, surprise, and fear.
2. Dimensional approach on the other hand is where we have emotions conceptualized based on their positions on a small number of dimensions i.e valence and arousal, which is what I’ll be using in this project as it helps reduce the dimension of training a MER model.

Valence refers to the space from negative to positive mood, and arousal is the space from calm to energetic. The paper (1809.07276) further explains that Arousal is highly correlated to audio while valence is highly correlated to both audio and lyrics. The project decision therefore, was to base the mood or emotion detection based on audio and lyrics.

For this project, I’ll be applying a concept in the paper (1909.05882) called 4Q, where emotional categories are further categorized in to 4 Quadrant Dimensions of Arousal and Valence as shown in the table below;

|  |  |  |
| --- | --- | --- |
| **Quadrant** | **Emotions** | **Synonyms** |
| Q1 (A+V+) | Joyful activation | joy |
| Power | - |
| Surprise | - |
| Q2 (A+V-) | Anger | angry |
| Fear | anguished |
| Tension | tense |
| Q3 (A-V-) | Bitterness | bitter |
| Sadness | sad |
| Q4 (A-V+) | Tenderness | gentle |
| Peace | - |
| Transcendence | spiritual |

# 2.3 DEEP LEARNING

Deep Learning is an extension of Machine Learning, that focuses on the use of multi-layer neural networks. These neural networks attempt to simulate the behavior of the human brain, allowing them to “learn” for large amounts of data.

Some Deep Learning architectures include Deep Neural Networks, Recurrent Neural Networks, and Convolution Neural Networks, and they have been applied widely in fields such as Computer Vision, Speech Recognition, and Natural Language Processing, among others.

Music Information Retrieval (MIR), and more specifically Music Emotion Recognition, since it involves the automatic classification of music, using Audio (can be represented as an image) and Lyrics (text), can apply Deep Learning in Image and Text Classification, which is used in this project. Some concepts on how this would be done is described next.

# 2.4 TRANSFER LEARNING

# 2.5 BIDIRECTIONAL ENCODER REPRESENTAIONS TRANSFORMER (BERT)

# CHAPTER 3 – METHODOLOGY

This section contains details on data collection and analysis, models architecture design decisions, training and evaluation, model deployment as an API, API design decisions, API testing and finally code refactoring.

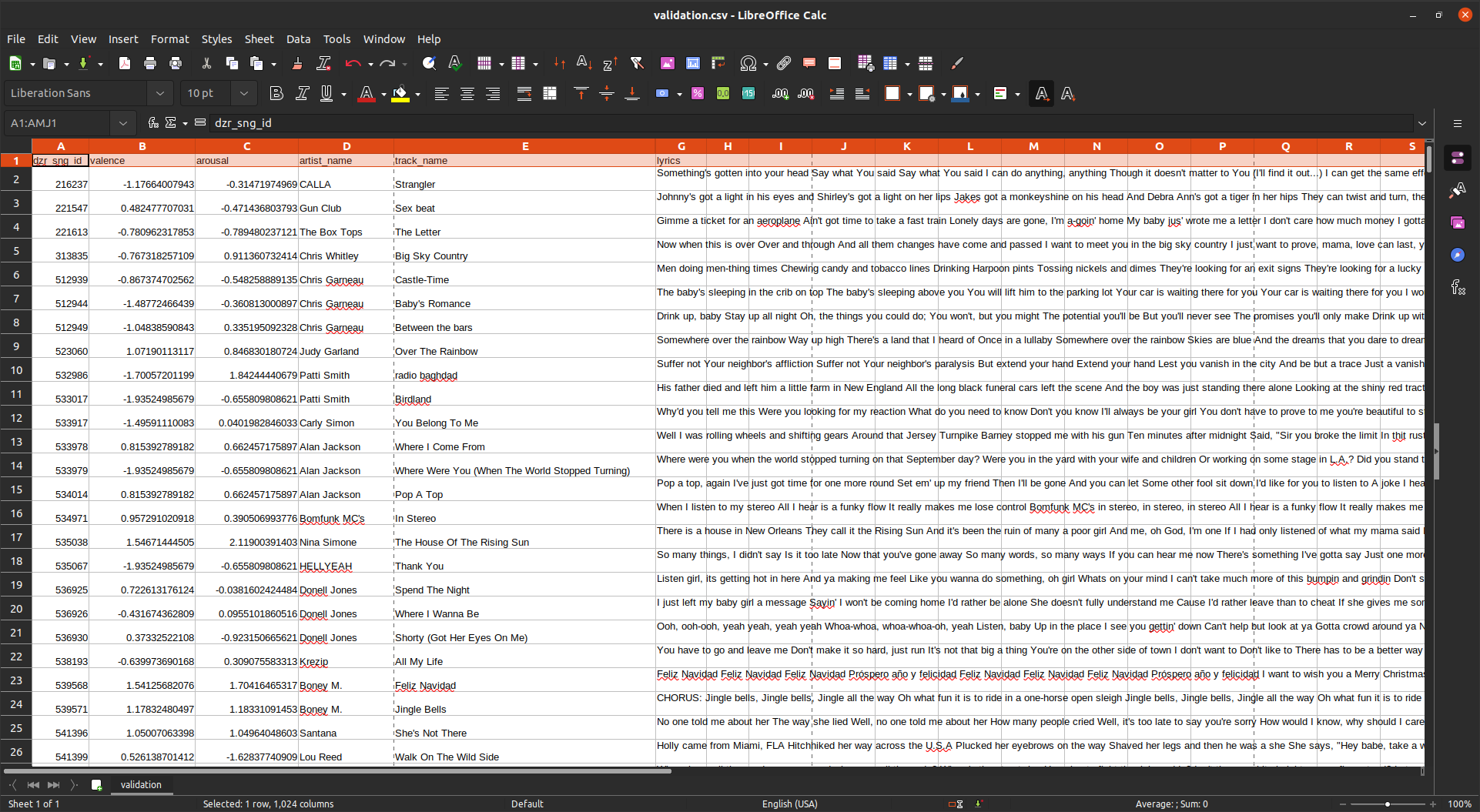
# 3.1 DATA COLLECTION AND ANALYSIS

The objectives on this sub-section includes; collection of lyrics, collection of song audio, setting the Valence and Arousal values into their respective *Quadrants*, described in the previous chapter, as the label column of our data, and finally cleaning the data.

The dataset selected for this project was the Deezer Music Emotion Recognition (MER) Dataset which was mentioned in (1809.07276) and was publicly available in the GitHub repository (<https://github.com/deezer/deezer_mood_detection_dataset.git>). The dataset came already split into training, validation, and testing sets, with data points 11000, 3000, and 3000 respectively. The dataset contained the following columns;

1. *dzr\_sng\_id,*
2. *MSD\_sng\_id,*
3. *MSD\_track\_id,*
4. *valence,*
5. *arousal,*
6. *artist\_name,*
7. *track\_name*.

A sample header of the dataset is as in the figure below;



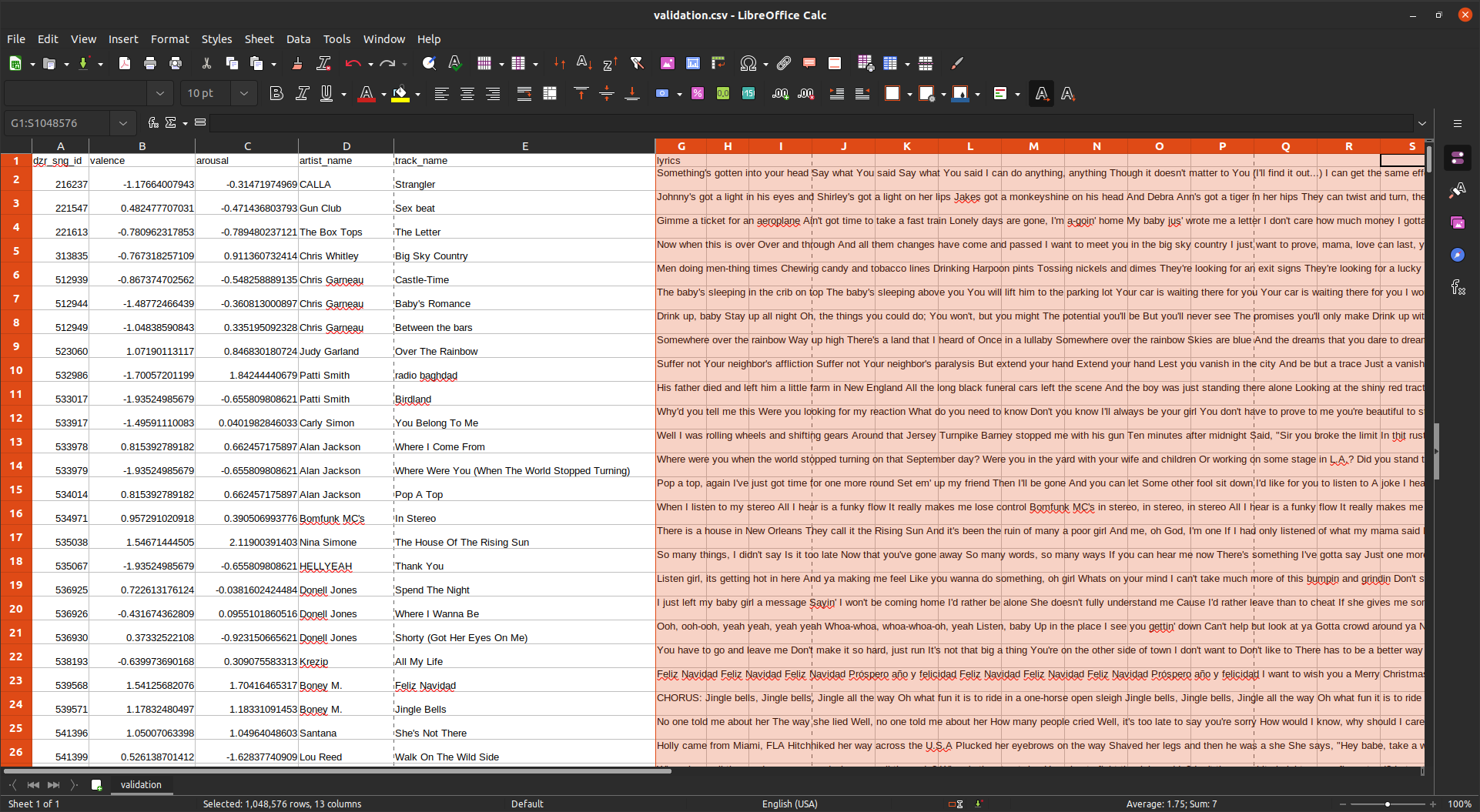
**Figure 3.1 Sample Dataset Header**

## 3.1.1 COLLECTION OF MUSIC LYRICS

In this particular task I needed to add a lyrics column to the dataset. The API selected for this task was the MusixMatch API, more specifically the lyrics endpoint. More details about the API can be found by following the following link to the API’s documentation <https://developer.musixmatch.com/documentation/api-reference/matcher-lyrics-get> .

Since I could not hit the endpoint and directly add lyrics to our dataset, I build a *lyrics\_service.py* program which could use query the endpoint using the *track\_name* and return the collected lyrics. These lyrics were then added to a new data feature/column named *lyrics* using the Numpy library.

A Sample dataset containing the *lyrics* feature/column is as in the figure below;



**Figure 3.2 Sample Training Dataset with Lyrics Feature/Column**

Since I was using the basic plan version of the MusixMatch API, I could only obtain 30% of a single song’s lyrics, which may be sufficient for the project.

## 3.1.2 COLLECTION OF SONG AUDIO

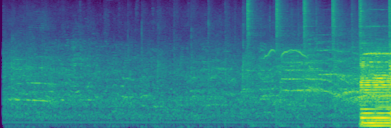
Originally, the plan I had for this sub-section was to use the Milliong Song Dataset (MSD) (<https://millionsongdataset.com/>) to get the songs audio due to the Deezer MER Dataset including MSD Track Ids for each datapoint. I although, ran into a problem of extracting the compressed audios from their HD5 format. Documentation from the MSD website included instructions in a wrapper Python file that I could have used, although it proved to be a challenge.

I settled for building my own program or service for downloading song audios. The workflow of the service included; being able to search for song using the song’s *track\_name* and *artist\_name*, downloading the music in .MP3 format, and finally converting the audio into a mel-spectrogram, a format used for audio classification in Deep Learning. The mel-spectrogram conversion was necessary since my audio model will be treating every song audio as an image. Libraries used included; YouTube-Search library responsible for finding a song’s YouTube link, YouTube-DL library for downloading the songs using the located link, and torchaudio (PyTorch’s audio processing library) and librosa for creating and saving the mel-spectrograms.

***The number of samples to get from an audio signal helped with cutting down or right padding the audio samples to a 30% length. I can use the average sampling rate of the songs in the dataset as base, then get the 30% from it.***

mel-spectrogram were used instead of regular spectrogram since the former is used in applications where we need to model human hearing perception, and audio classification applications, which is necessary in this project.

Sample mel spectrogram is as shown in the figure below;



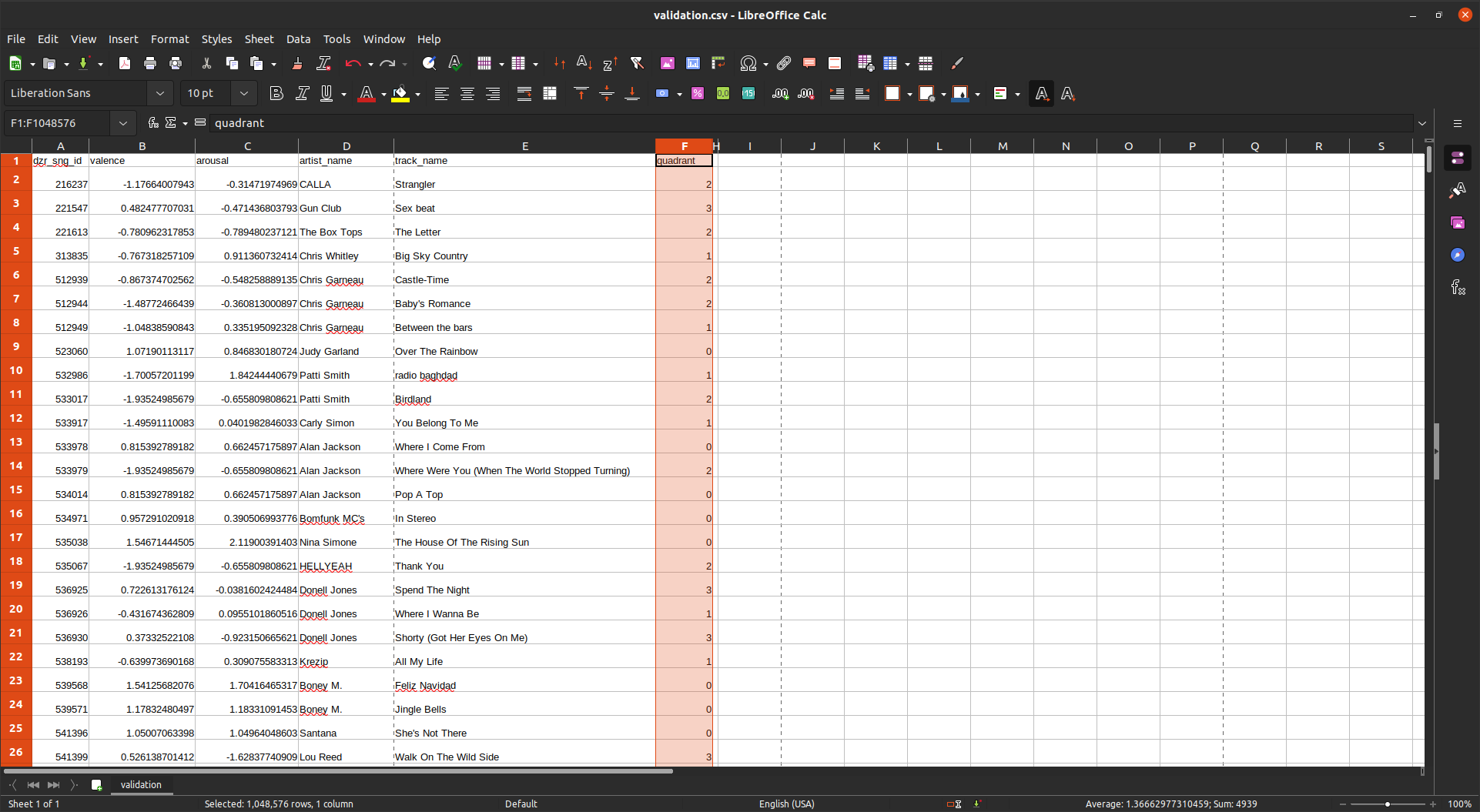
**Figure 3.3 Sample Mel-Spectrogram File**

Due to copyright reasons, all song audio files are deleted after their spectrograms have been saved. The spectrograms were all saved in using their *dzr\_sng\_id* value.

## 3.1.3 QUADRANTS AS LABELS

As shown in **Figure 3.1,** the dataset contains *valence* and *arousal* features/columns. These values can either be positive or negative values to match the *Quadrant* divisions, when combined, as described in the previous chapter.

I created a new data label feature/column called *quadrant*, which contained integer-values ***i*** of range ***0* *to 3*** representing the ***Qi*** a datapoint belongs to, such that ***Qi = 0*** *==* ***Q1***… ***Qi=3*** *==* ***Q4.*** All this was done using a function utilizing the Pandas libraryand sample results is as in the figure below;



**Figure 3.4 Sample Dataset with Quadrant Feature/Column**

## 3.1.4 CLEANING DATA

For this process, I deleted all columns that were no longer necessary to train the models, which included; *dzr\_sng\_id, MSD\_sng\_id, MSD\_track\_id, valence, arousal, artist\_name* and *track\_name*. Other datapoints containing missing or Nan values were also dropped.

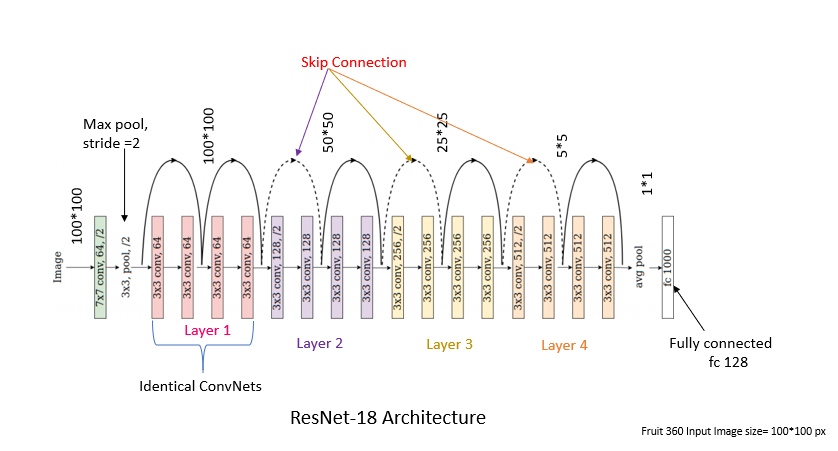
In the lyrics feature column, the lyrics datapoints contained the text **“*\*\*\*\*\*\*\* This Lyrics is NOT for Commercial use \*\*\*\*\*\*\* (1409621065547)*”,** at the end of every song lyrics, which I considered as unnecessary information that might lead to some skewedness in the data, so I trimmed it.

# 3.2 MODEL ARCHITECTURES, TRAINING AND EVALUATION

As described in the paper by (ref), there are two ways in which Music Emotion Recognition can be accomplished using Deep Learning, one is having two models, one for lyrics and the other for audio only, and two having a combined version of the two. They described that having the combined version seemed to have a better performance than separate models. For my algorithm, I decided to use the two models approach as it would help in faster debugging and development, and also help in improving on the results of the models described in the paper by aggregating the outputs of the two models. This also acts as a form of regularization of the generated results.

## 3.2.1 AUDIO MODEL

The audio model is a finetuned version of the ResNet18 model architecture described in the transfer learning section. The architecture is as shown in the figure below;



**Figure 3.3 Audio Model Architecture**

Since the model expects input to be images (the 2-Dimensional spectrograms), created in the last section, this task can be considered an image classification task in Computer Vision. I then applied transfer learning on the ResNet18 model.

During training, the optimization function selected was the Adam and criterion or loss function used was the Cross Entropy Loss as they perform really well for multi-class classification tasks like this one. The training loss and accuracy score was used to monitor the model’s performance on the validation set. The graph of training losses over the 150 epochs training time, is as shown in the figure below;

**Figure XYZ Training Losses Over 150 Epochs**

As for evaluation or testing I used model accuracy scores per class. The results were as shown in the figure below;

**Figure XYZ Accuracy Scores per Class**

## 3.2.2 LYRICS MODEL

The architecture for the lyrics model is as in the image below.

(architecture)

Since the lyrics task is a text or sequential classification task in Natural Language Processing (NLP), for this model, we used finetuned a Bidirectional Encoder Representations Transformer (BERT) model, described in the literature review section. More specifically the *bert-base-uncased* pre-trained model from the transformers library, as we assumed that for example, the word BAD and bad, both shared the same sentiment and since the *base* model only had 110 million parameters, which is I could train locally on my computer.

The entire corpus of words was tokenized or encoded using transformers’ BertTokenizer which is based on WordPiece, a subword segmentation algorithm used in natural language processing. The vocabulary is initialized with individual characters in the language, then the most frequent combinations of symbols in the vocabulary are iteratively added to the vocabulary.

(WordPiece as used in BERT Image)

During training, the optimization function selected was Adam similar to the audio model, and loss metric was used to measure the models performance. Weighted F1 score was also used, as it tells you how precise your model is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances). The graphs for both loss and F1 score over the 5 epochs training time are as follows;

(images)

(images)

As for evaluation or testing I used model accuracy scores per class. The results were as follows;

(image)

# 3.3 SOFTWARE REQUIREMENTS

***Get main software requirements from the requirements.txt file.***

# 3.4 HARDWARE REQUIREMENTS

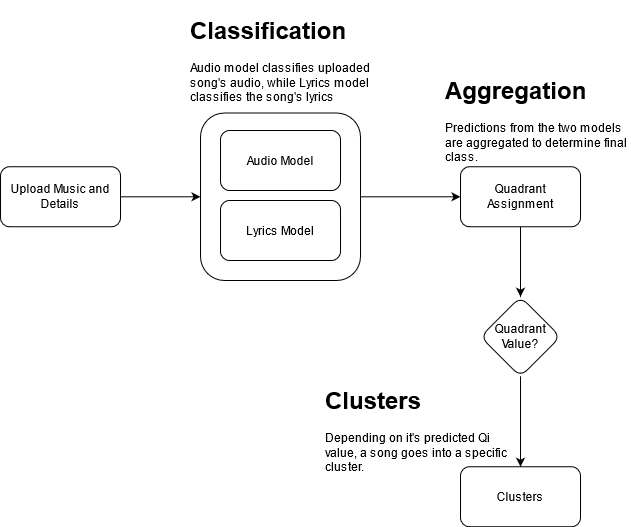
1. NVIDIA 1660 GPU, version 470 CUDA 11.0, RAM 6 GB

2. AMD RYZEN 3600 7 16 CORES

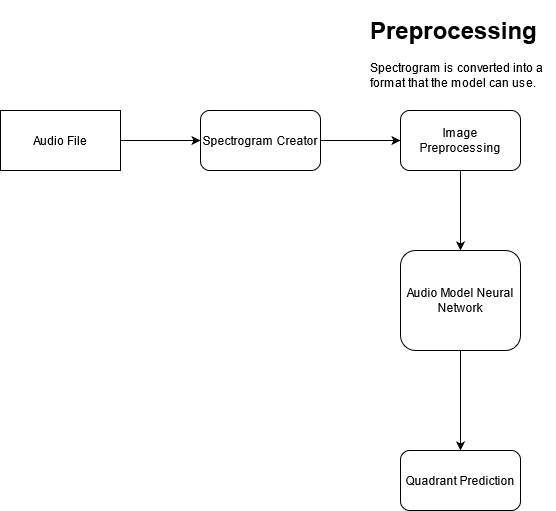
3. 16 GB DDR3 RAM

# CHAPTER 4 – PROJECT PRESENTATION

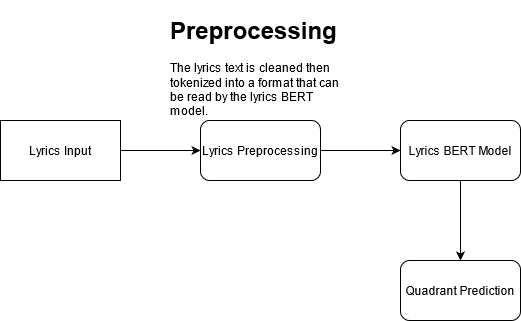
# 4.1 DATA FLOW DIAGRAMS (DFDs)



**Figure 4.0 Clustering Pipeline**



**Figure 4.1 Audio Model Pipeline**



**Figure 4.2 Lyrics Model Pipeline**

# 4.2 PROJECT SCREENSHOTS

# CHAPTER 5 – RECOMMENDATIONS AND CONCLUSION