Software Architecture Document

Music Analysis System

Version 3

25/07/2021

Table of Contents

Software Architecture Document

1. Introduction

Music Analysis system is a platform where an audio file would be given to the system and various music labels in a recording like notes, number of instruments used, composer, onset and the next note would be predicted by the system. This document elaborates the software architecture for the Music Analysis System. The system architecture has many components which are explained in detail.

1.1. Purpose

The goal of our project is to build a Music Analysis System that can predict music labels in a recording like notes, number of instruments used, composer, onset and the next note.

1.2. Scope

AI technology has the ability to learn a musician's style and sound by analyzing previous songs, common note combinations, and the creator's likely next move. So this project will help analyze instruments, composers, and notes to determine playlists, artists, and songs that fit the style of the user.

1.3. Definitions, Acronyms, and Abbreviations

- **LSTM** - Long Short Term Memory
- **EDA** - Exploratory Data Analysis
- **GRU** - Gated Recurrent Unit
- **AWS** - Amazon Web Services
- **CNN** - Convolutional Neural Networks
- **IDE** - Integrated Development Environment
- **CSV** - Comma Separated Version
- **DFD** - Data Flow Diagram
- **FFT** - Fast Fourier Transformation
- **CQT** - Const Q Transformation

1.4. References

[1]https://www.kdnuggets.com/2020/02/audio-data-analysis-deep-learning-python-part-1.html
[2]https://rua.ua.es/dspace/bitstream/10045/76991/1/Automatic_music_transcription_using_neural_networks_MINGUEZ_CARRETERO_MANUEL.pdf
[3]https://medium.com/@ianvonseggern/note-recognition-in-python-c2020d0dae24
[4]https://analyticsindiamag.com/7-python-libraries-for-manipulating-audio-that-data-scientists-use/
[5]https://www.analyticsvidhya.com/blog/2017/08/audio-voice-processing-deep-learning/

1.5. Overview

In order to fully document all the aspects of the architecture, the Software Architecture Document contains the use case view, describes the architectural goals and constraints of the system, and architectural representation.

Section 2: describes the use of each view

Section 3: describes the architectural goals and constraints of the system

Section 4: describes the most important use-case realizations

Section 5: describes logical view of the system including interface and operation definitions.

Section 6: describes the data view of the system

2. Architectural Representation

The views used to document the Music analysis system are:-

Use Case view

Audience: Software Engineer, Music Analyst
Area: The music labels in a recording like notes, number of instruments used, composer, onset and the next note would be predicted by the system. These labels can be used by various music streaming services to determine the quality of a song and plan their strategy.
Related Artifacts : Use-Case Diagram

Logical view

Audience: Music Analysts
Area: Users can give an audio segment as the input and the algorithm must predict the required labels associated with different time stamps for that audio.
Related Artifacts: Activity Diagram

Data view

Audience: Data Specialists, Database Administrators
Area: We have a data of 330 music recordings. This data can be saved on a server or or on a local machine. The algorithm can dynamically fetch the database during run time.

3. Architectural Goals and Constraints

There are some key requirements and system constraints that have a significant bearing on the architecture. They are –

1. The size of the dataset is unusually large, which is why it is extremely necessary to use a high end virtual machine with commendable graphics and RAM.

2. The program will be written in python programming language using various in-built modules. These modules give a certain edge to perform pre-processing and analysis to the user which makes the learning process efficient and effective.

3. The primary goal of the system will be able to analyze the music recordings to identify and to predict various labels for that recording. Thus, a system predicting the music recording's notes, identifying composers, onsets would be very useful for the music analyst and music streaming companies to analyse the song quality.

4. Under any circumstances, if the system cannot be deployed on cloud platforms then we will have to perform the analysis on *Google Colab* or *Jupyter Notebook*. But, for this we might need to consider taking just 10-15 (maybe less than this) audio files as the pre-processing alone is a time consuming process in our case.

5. Given the pre-processing is complete using FFT or CQT methods of spectrogram generation, one cannot ignore the fact that the model training may pose challenges because of less specifications of the personal computer.

4. Use-Case View

4.1. Actors

There will be two end users associated with this project.

i) Engineer - The engineer will be responsible for all the background operations involved in the working of the system. These operations include data updation at regular intervals and keeping the performance of the algorithm in check.

ii) Music Analyst - The music analyst will be able to predict the instruments used in a particular classical music segment.

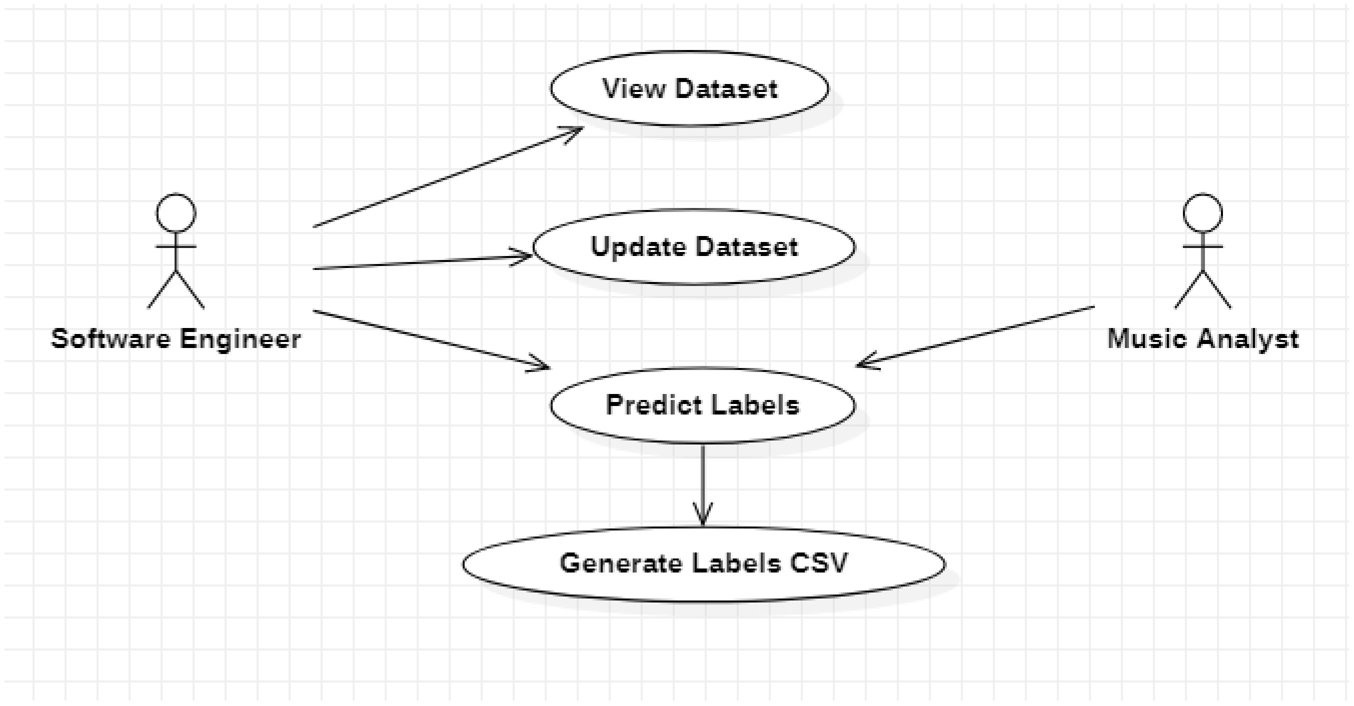

Figure - 1 Use Case Diagram for Music Analysis System

4.2. Use Cases

4.2.1 View / Manage Dataset

This authority lies in the hands of the engineer. Not only can they view the data on the servers (or any local system) but can also update new entries so that the model is trained with new inputs on a regular basis. The data fed by them will be saved in the software's database so that it can be used for further training of the algorithm.

4.2.2 Predict Labels

The system allows both the users to predict the labels for a song clip saved as a *.wav* file and generate the report.

Note: The input audio will be rejected if it doesn't clear the error check i.e., if it is blank, corrupted or saved with incorrect format.

5. Logical View

i) Firstly, we will begin with the visualisation of the audio signals to determine and eliminate outliers. In this step, we will analyse audio files for discrepancies like blank file or a zero KB file or a corrupted file and will reject it. It will also reject the files which are of any format other than .wav or saved by the .wav extension. This step will be performed using the *ffmpeg* function in Bash Script.

ii) The second step would be preprocessing. The primary goal was to perform this step by converting *time domain features* of the audio to *frequency domain features* so that they can be further converted into spectrograms. But because of limited resources, we will use the traditional spectrogram conversion method to prepare the data for model training as both FFT and CQT methods of audio pre-processing require high end graphics unit to study audio features of certain timestamps.

iii) These *mel spectrograms* will be further converted into images for model training. The initial goal was to use spectrograms to study time domain features to predict the instruments, onsets, start beat, end beat, etc. But now we will use image processing to predict only the type of instruments used in the audio.

iv) We will use LSTM, one of the most renowned algorithms in deep learning for image data training, to build the model. The model will have an *Attention Mechanism* architecture following a bi-directional approach.

v) This model will, then, be tested on our test data, which will have some classical audio files along with audios of other genres. We will also test it with short and long duration inputs as well.

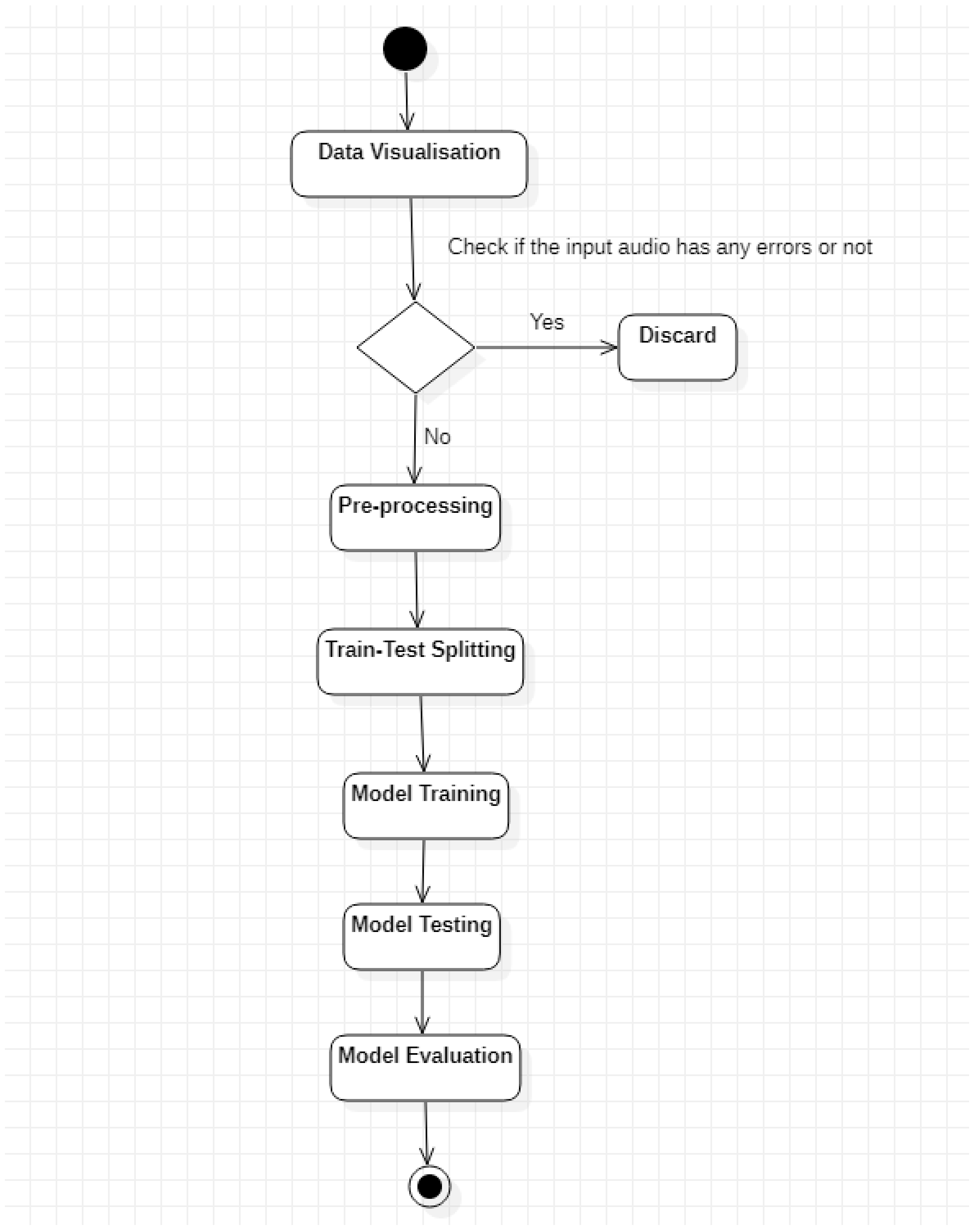

Figure - 2 Activity Diagram for the Labels Prediction Model

6. Data View

The below figure shows the flow of data through the various stages of the music analysis system

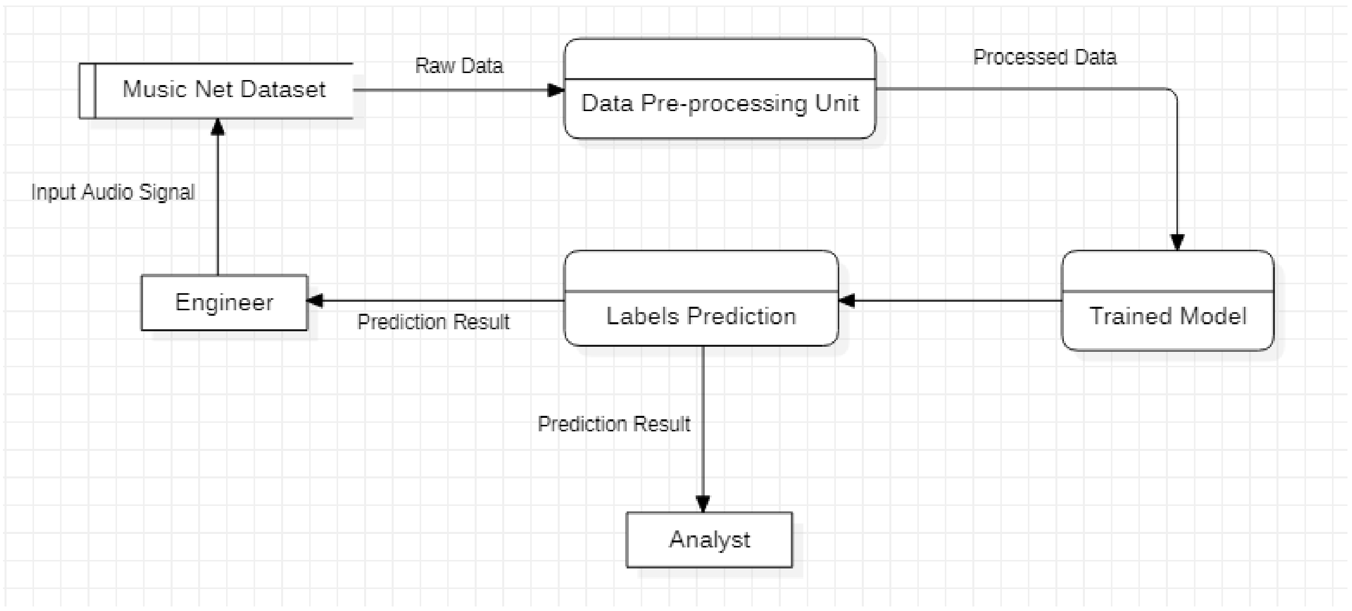

Figure - 3 Level - 1 DFD for Music Analysis System

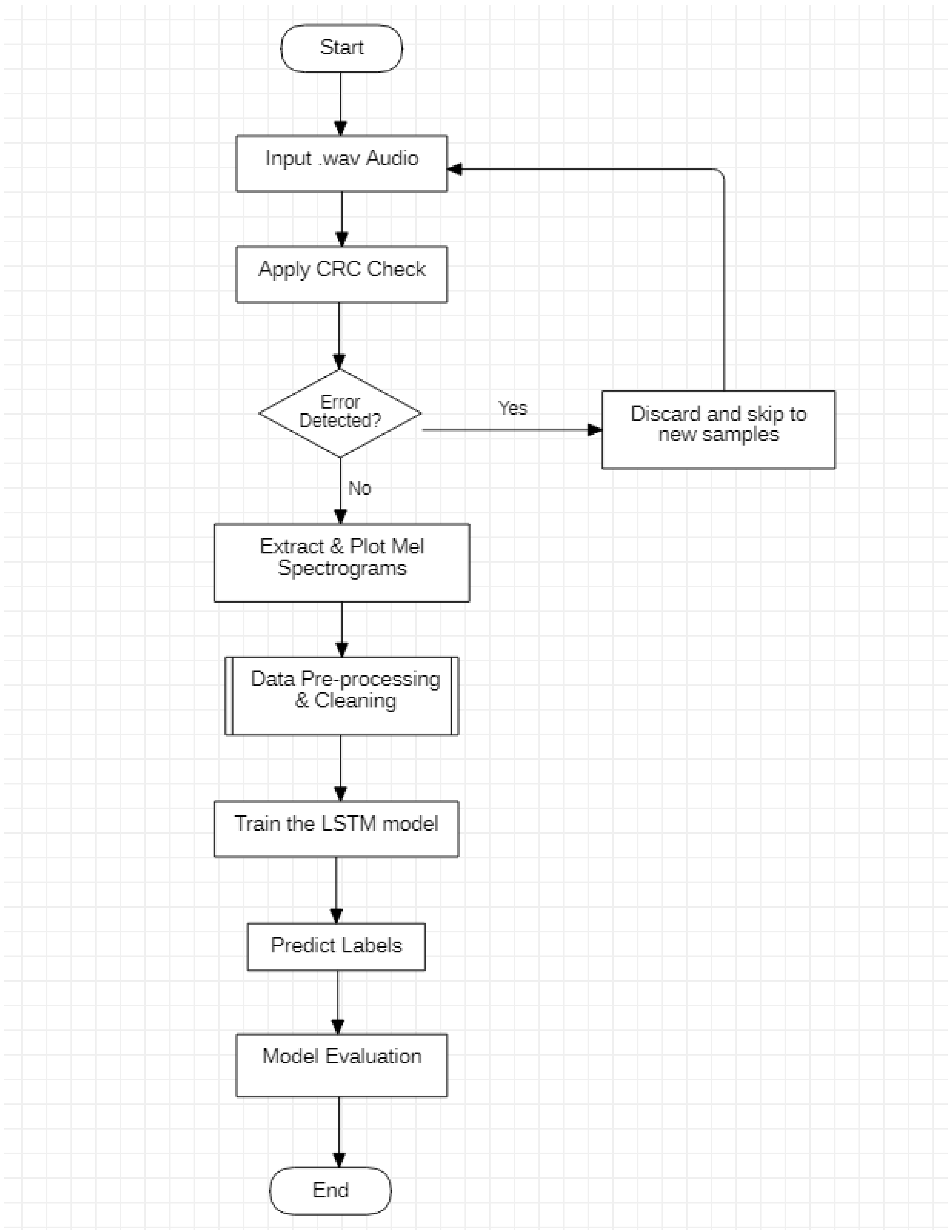

Figure - 4 Flow Chart for Music Analysis System

7. Future Scope

The proposed model will be able to predict the type of instruments used to compose the classical audio. But considering the abundance and flexibility of the data, whenever high-end resources are available to the team, this project can be further used to predict the time dependent features of the music like start beat, end beat, onsets, etc.