S104: ICCM: Instrument Classification in Carnatic Music

Project Plan, [version 2.0]

Guillem Gauchia, Àlex Herrero, Gerard San Miguel, Roddie Mc Guinness
11/05/2023

Introductions

The purpose of this project is to develop a classification system that can identify and classify the different instruments used in Carnatic music. Our system will use machine learning algorithms to analyze audio recordings of Carnatic songs and classify the different instruments used in them.

Carnatic music is a classical form of music originating from South India, with a rich tradition spanning over several centuries. It is a melodic and rhythmic system that follows a structured framework with a set of ragas, talas, and compositions. The music is characterized by intricate melodies, intricate rhythms, and improvisation.

Carnatic music is performed using a variety of instruments, with some being more predominant than others. The most prominent instruments include the human voice, veena, violin, flute, mridangam, tambura, and ghatam. Other instruments such as the tabla and kanjira are also used, but to a lesser extent.

As a group, we agreed with our professor during the early stages of the course to find a project theme relating to Carnatic music that aligns with our interests and abilities. After some discussion and refinement, we settled on the idea of developing a classification tool that can accurately identify and track the instruments used in Carnatic music performances. We will train this tool with appropriate machine learning techniques. In particular, the tool will be capable of producing time stamps of when each instrument enters, exits, and re-enters in a given performance.

We think that this will be a valuable tool for musicologists, researchers, and musicians who are interested in understanding the intricacies of Carnatic music and its instrumentation.

Team responsibilities

Effective collaboration within a diverse group requires careful consideration of the unique strengths and areas of expertise of each team member.

Given that our group is diverse in terms of university courses, some members may be more skilled in certain areas of the project than others. An important aspect to highlight is that, in terms of music studies, certain members of the group are much more qualified than others, while in software development, other members may have more capabilities due to their previous academic knowledge.

Therefore, to make the most of our diverse skill sets, we have tried to adapt the organization and distribution of responsibilities to best fit the qualities of each student. As such, we have assigned tasks to the members with the most relevant expertise and knowledge. These task leaders will oversee the work of the team, and ensure a high standard is produced, but the entire group will work collaboratively to ensure that the project is successful.

Document Deliverables:

- **Individual Weekly Report:** Each member will be working individually to fill the weekly report before every class, in order to discuss at class what every member of the group has been working on.
- **Project Plan:** Àlex Herrero, Guillem Gauchia and Roddie Mc Guinness
- **State Of the Art:** Roddie Mc Guinness
- **Software Development Tools:** Àlex Herrero and Guillem Gauchia
- **Software Requirements Specification:** Àlex Herrero and Guillem Gauchia
- **Ethical Considerations:** Oriol Roca and Gerard San Miguel
- **Software Evaluation:** Roddie Mc Guinness,Oriol Roca and Gerard San Miguel
- **Final Presentation:** All the members of the group will be responsible for carrying out their part of the presentation.

Project Development:

- **Data:** Àlex Herrero Díaz and Guillem Gauchia
- **Data Processing:** *To be determined*
- **Feature Extraction:** Oriol Roca and Gerard San Miguel
- **Characterization:** Roddie Mc Guinness
- **Modeling:** *To be determined*
- **Evaluation:** Roddie Mc Guinness
- **Annotation:** Àlex Herrero and Guillem Gauchia

Integration and Quality of the Project:

- **Good programming practices:** Àlex Herrero
- **Organize group meetings:** Àlex Herrero and Gerard San Miguel
- **Github repository management:** Gerard San Miguel
- **Google Drive organization:** Guillem Gauchia

Schedule

Marked in red represents the report deliverables and when the group will work on each of the documents, and marked in blue is an indicative guide on when the group will be working on each part of the project.

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10
INDIVIDUAL WEEKLY REPORT										
PROJECT PLAN										
STATE OF THE ART										
SOFTWARE DEVELOPMENT TOOLS										
SOFTWARE REQUIREMENTS SPECIFICATION										
ETHICAL CONSIDERATIONS										
SOFTWARE EVALUATION										
FINAL PRESENTATION										
DATA										
DATA PROCESSING										
FEATURE EXTRACTION										
CHARACTERIZATION										
MODELING										
EVALUATION										
ANNOTATION										

Risk management

As with any project, it is important to anticipate and manage potential risks that could diminish the quality of the end result. This section serves as a starting point for this process, in foreseeing and planning for various risks we may face.

Several factors could impede our progress and hinder the effectiveness and completeness of our final product. As such, it is important to foresee, as much as possible, any and all potential impediments. We have categorized these potential risks into two main categories: Technical and Software-related risks, Communication and Team-related risks. Below we discuss an extensive list of the risks in each category. It is important for us to understand that this is list is not exhaustive, as unforeseen or overlooked issues often arise in projects such as these

Technical and Software-related risks

- Time constraints on the training and tuning of our instrument classification model:
    - Solution: Ensure that this portion of the project is started on time, and that the training is closely monitored. Furthermore, it is important that we accurately track the different results achieved through the altering of various parameters as we tune the model.
- Differing levels of experience in machine learning and modeling
    - Solution: Work closely with Thomas to ensure that our understanding of new topics is accurate.
- Other general implementation issues and challenges
    - Solution: Discuss issues with implementation as they arise, with both Thomas and the group. Furthermore, aim for simple and clean code as opposed to intricate solutions.
- Difficulty differentiating between instruments with similar sound profiles
    - Solution: May involve lots of tuning of the model parameters, which could be time-consuming, and feeds into the above consideration on model-tuning.
- Noise, leakage, and generally poor quality audio
    - Solution: The data cleaning and processing stage will be crucial to deal with this issue.

Communication and Team-related risks

- Poor concurrent-development practices, poor use of Github
    - Solution: Manager of Github repository has been appointed to ensure the repo is maintained and used correctly, and that branch merges, pushes, and pulls are all utilized effectively.
- Miscommunication and misunderstanding between team members
    - Solution: To prevent miscommunication and ensure we're all on the same page, we have established regular check-ins using various communication tools such as Slack, and WhatsApp.
- Poor workload distribution

    - Solution: In order to avoid certain members of the group having a greater workload, or getting stuck in one part of the project leading to carrying an unsustainable workload, the members of the group must communicate the status of the work they are doing.
- Differing levels of experience in Signal Processing and Music Knowledge
    - Solution: As certain members of the group start from prior knowledge on the matter, these members should take responsibility for the aspects that are required and offer guidance on the matter to those members who need it.

To reiterate, this is a non-exhaustive list, and it is crucial for each member to remain vigilant and proactive in identifying and mitigating other issues that may arise during the project.

S104: Instrumentation Classification in Carnatic Music

State of the art, version 2

Guillem Gauchia, Àlex Herrero, Gerard San Miguel, Roddie Mc Guinness

28/05/2023

Introduction

In this document, we review the current state of the literature relating to our chosen project and application. We describe and give an overview of the most relevant publications in the Literature Review section. Following this, in the Technology Review section, summaries are given of the most commonly and successfully used techniques for each of the main stages of our pipeline, Feature Extraction and Selection, Modelling, and Evaluation. At the end of each of these sections, we conclude with our choices for each of these stages. Finally, in the Application Review section, some of the applications currently available which deal with some of the processes we will be working on are also summarized.

Literature review

This section aims to provide a comprehensive review of instrument classification and characterisation techniques applied to Indian Classical Music, and specifically to Carnatic music. The scope of this review includes the description of various papers on the classification and characterisation of musical instruments in the context of the music. The section will focus on recent research published within the past five years, and covers a range of sources.

“Group delay based music source separation using deep recurrent neural networks” studies Modified Group Delay for learning time-frequency masks of audio sources, i.e., instruments. This is demonstrated in two different tasks, the second of which is of particular interest to us: vocal-violin separation on a Carnatic music data set. It is found in the paper that this feature improves results on various models with respect to the Signal to Interference Ratio. (1)

“Indian Instrument Identification from Polyphonic Audio using KNN Classifier” uses a K-Nearest Neighbour algorithm to identify 7 different Indian instruments, including the violin, in the context of polyphonic music. This study uses 34 features to achieve an F1 measure of 97.3%. (2)

While the main focus of the study, “On the application of deep learning and multifractal techniques to classify emotions and instruments using Indian Classical Music”, is not of huge significance to us, a secondary task in this study is to classify instruments in Indian classical music. Convolutional Neural Network based architectures were used in this study to differentiate between the sitar and the sarod - two primarily Hindustani, plucked, string instruments. This task is completed with close to perfect accuracy, there were zero misclassifications of sitar clips, and just three out of two hundred sarod clips were misclassified. (3)

Another paper not strictly falling within the scope defined earlier, but which may prove useful, is “Classification of Musical Instruments using SVM and kNN”. This paper examines the performance of SVM and kNN algorithms in classifying 16 different instruments using MFCCs and Sonograms. The combination of MFCC and Sonogram in the kNN model produced 98% accuracy, while the same combination in the SVM system produced 99% accuracy. (4)

“Content-Based Music Information Retrieval (CB-MIR)and Its Applications toward the Music Industry: A Review”, analyzes many facets of music information retrieval, of which we have a particular interest in two areas. These two are section 4, which deals with Vocal/Non-Vocal Segmentation, and section 10, which deals with Instrument Identification. Section 10 suggests that very few of the most relevant and well-cited papers on the topic use raw data clips, and more often than not, clips of non-overlapping instruments have been used. This could pose challenges for us in adapting to our dataset which does contain tracks with overlapping instruments. (5)

“Feature Selection and Classification of Indian Musical String Instruments Using SVM” discusses how different feature selection strategies can affect a SVM model for Indian string instruments. In all three experiments, accuracy upwards of 92% was achieved; the best results were achieved using the top 5 audio features extracted with MFCC using an SVM classifier. (6)

Technology review

From the above literature review, we have found that there are a number of different techniques and tools that could be used in the various steps of our project.

Firstly, in the feature extraction phase, there are a wide variety of features to consider. MFC coefficients have been used in many instrument classification papers and studies. MFC - Mel-frequency cepstrum is a frequency spectrum investigation tool, used in particular for investigating periodic patterns. It has differing definitions, but usually refers to the inverse Fourier transform of the log of the frequency spectrum, mapped to the mel-scale. The coefficients which make up an MFC are known as MFC coefficients (MFCCs). Sonogram, or spectogram, is another feature that has been used by others in instrument classification, this is a graphical depiction that illustrates how the frequency spectrum of a signal changes over time. Standard statistics such as mean, standard deviation, max and min, etc are also used. LPC - linear predictive coding - is another commonly extracted feature for this application. LPC uses a linear predictive model to estimate and represent the spectral envelope of an audio signal. Many other features have been used also, such as zero crossing rate - the number of times a signal changes sign, spectral bandwidth - described by (7) as “the weighted average of the frequency signal by its spectrum”, and energy. As mentioned, another interesting feature which has been used is modified group delay. (2,4,6–12)

The features we have chosen for our model are MFCC, spectral centroid, spectral flatness, spectral bandwidth, band energy ratio, amplitude envelope, root mean squared energy, zero crossing rate, maximum, minimum, mean, and standard deviation.

In the feature selection process, different studies have used a broad range of tactics. Some studies, such as (2) have used all the features extracted in their models, while others, such as (6) have used systems such as multivariate analysis of variance and chi-squared test to choose a small number of the best features for their models. In our study, we will take the former approach and use all of the features we extract.

Coming to the model itself, the most commonly used approaches have been the use of kNN classification and SVM classification. In the context of our task, a kNN algorithm would involve using a distance function with features of a particular datapoint as input, and outputting whether that datapoint is a member of the class based on the k most similar previously classified datapoints. SVM on the other hand is a type of supervised learning model in which classification is done by constructing an n-1 hyperplane in the space of n characteristics being used, points are then classified based on which side of the hyperplane they lie. Another model that has also been used is Convolutional Neural Networks (CNN). A CNN, according to (13,14), is a type of feedforward neural network which uses mathematical convolution in hidden layers, and which does not require manual feature extraction. (2,4,6,7,15,16) The model we have chosen for our case is gradient boosted classification trees; we noticed its seeming absence in tasks relating to Indian instrument classification, and so this should be a relatively novel approach for this class of problem.

In evaluating a classification model, some of the most commonly used metrics and techniques include precision, recall, F1 score, accuracy and confusion matrices. Precision refers to the fraction of total positive classifications that are true positives, (17) so letting TP = True Positives, and FP = False Positives we have:

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

Recall refers to the fraction of total number of positive datapoints which are classified as positive, (17) i.e., letting FN = False Negatives we have:

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

F1-score is defined by Wikipedia as “the harmonic mean of precision and recall, (18) i.e.

$$F1 = \frac{2}{Recall^{-1} + Precision^{-1}} = 2\frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}$$

Accuracy is the number of correct choices over the total number of choices, (19) which, setting TN = True Negatives, gives us:

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

Finally, the confusion matrix is a way of visualizing class prediction versus true class for each different class. (20) In our case, the confusion matrix for violin classification would look something like the below:

Total = P + N = PP + PN	Predicted Positive (PP)	Predicted Negative (PN)
Actual Positive (P)	True Positive	False Negative
Actual Negative (N)	False Positive	True Negative

In evaluating our model, we are going to use precision, recall, F1, and accuracy.

Applications review

There are a number of audio separation services available for commercial use currently, some examples include:

- Lalal.ai: an AI-powered program for splitting voices and instruments from mixed audio tracks. This application uses neural networks to split files, extract vocals, and more while minimizing sound artifacts. (21)
- AudioSourceRE: offers a number of different AI-based vocal and instrumental separation solutions, alongside their other commercial audio production tools. (22)

Another interesting application on the market is “intelliScore”. This application takes an audio file and converts it into a MIDI file separating the audio into its different component instruments and their notes.

As of now, there are no applications on the market for annotating music tracks with instrument timestamps, our application will accomplish this task in the specific case of Carnatic music.

Apart from the above, there are a number of different software libraries which are useful for development of a wide range of audio processing applications such as the ones listed above and the one we intend to develop. Two of these are librosa and essentia. Librosa is a python library with tools for audio processing and analysis, and which “provides the building blocks necessary to create music information retrieval systems”.(23) Essentia is a C++ library with a python wrapper that includes a wide variety of tools for audio analysis including “flexible and easily extendable algorithms for common audio analysis processes and audio and music descriptors”.(24)

References

1. Sebastian J, Murthy HA. Group delay based music source separation using deep recurrent neural networks. In: 2016 International Conference on Signal Processing and Communications (SPCOM). 2016. p. 1–5.
2. Chandan SV, Naik MR, Ashwini, Krishna AV. Indian Instrument Identification from Polyphonic Audio using KNN Classifier. In: 2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET). 2019. p. 135–9.
3. Nag S, Basu M, Sanyal S, Banerjee A, Ghosh D. On the application of deep learning and multifractal techniques to classify emotions and instruments using Indian Classical Music. Physica A: Statistical Mechanics and its Applications. 2022 Jul 1;597:127261.
4. Rathikarani V, Dhanalakshmi P, S. P. Classification of Musical Instruments using SVM and KNN. International Journal of Innovative Technology and Exploring Engineering. 2020 May 1;9:2278–3075.
5. Murthy YVS, Koolagudi SG. Content-Based Music Information Retrieval (CB-MIR) and Its Applications toward the Music Industry: A Review. ACM Comput Surv. 2018 Jun 12;51(3):45:1-45:46.
6. Chaudhary SR, Kakarwal SN, Bagade JV. FEATURE SELECTION AND CLASSIFICATION OF INDIAN MUSICAL STRING INSTRUMENTS USING SVM. INDJCSE. 2021 Aug 20;12(4):859–67.
7. Racharla K, Kumar V, Jayant CB, Khairkar A, Harish P. Predominant Musical Instrument Classification based on Spectral Features. In: 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN). 2020. p. 617–22.
8. Mel-frequency cepstrum. In: Wikipedia [Internet]. 2023 [cited 2023 May 7]. Available from: https://en.wikipedia.org/w/index.php?title=Mel-frequency_cepstrum&oldid=1152665449
9. Spectrogram. In: Wikipedia [Internet]. 2023 [cited 2023 May 8]. Available from: https://en.wikipedia.org/w/index.php?title=Spectrogram&oldid=1139424862
10. Linear predictive coding. In: Wikipedia [Internet]. 2023 [cited 2023 May 9]. Available from: https://en.wikipedia.org/w/index.php?title=Linear_predictive_coding&oldid=1152198549
11. Zero-crossing rate. In: Wikipedia [Internet]. 2021 [cited 2023 May 9]. Available from: https://en.wikipedia.org/w/index.php?title=Zero-crossing_rate&oldid=1020703960
12. Shetty S, Hegde S. Automatic Classification of Carnatic Music Instruments Using MFCC and LPC. In: Sharma N, Chakrabarti A, Balas VE, editors. Data Management, Analytics and Innovation. Singapore: Springer; 2020. p. 463–74. (Advances in Intelligent Systems and Computing).
13. Li Z, Liu F, Yang W, Peng S, Zhou J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. IEEE Transactions on Neural Networks and Learning Systems. 2022 Dec;33(12):6999–7019.
14. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. Pattern Recognition. 2018 May 1;77:354–77.
15. *k*-nearest neighbors algorithm. In: Wikipedia [Internet]. 2023 [cited 2023 May 9]. Available from: https://en.wikipedia.org/w/index.php?title=K-nearest_neighbors_algorithm&oldid=1153790564
16. Support vector machine. In: Wikipedia [Internet]. 2023 [cited 2023 May 9]. Available from: https://en.wikipedia.org/w/index.php?title=Support_vector_machine&oldid=1144271534
17. Precision and recall. In: Wikipedia [Internet]. 2023 [cited 2023 May 9]. Available from: https://en.wikipedia.org/w/index.php?title=Precision_and_recall&oldid=1149017180
18. F-score. In: Wikipedia [Internet]. 2023 [cited 2023 May 9]. Available from: https://en.wikipedia.org/w/index.php?title=F-score&oldid=1148225663
19. Accuracy and precision. In: Wikipedia [Internet]. 2023 [cited 2023 May 9]. Available from:

https://en.wikipedia.org/w/index.php?title=Accuracy_and_precision&oldid=1139395873
20. Confusion matrix. In: Wikipedia [Internet]. 2023 [cited 2023 May 9]. Available from: https://en.wikipedia.org/w/index.php?title=Confusion_matrix&oldid=1148699071
21. GMBH O. Vocal Remover & Instrumental AI Splitter | LALAL.AI [Internet]. [cited 2023 May 9]. Available from: https://www.lalal.ai/
22. Developer PH. AudioSourceRE Vocal & Instrumental Isolation Software [Internet]. [cited 2023 May 9]. Available from: https://www.audiosourcere.com/home
23. librosa — librosa 0.10.0 documentation [Internet]. [cited 2023 May 9]. Available from: https://librosa.org/doc/latest/index.html
24. Homepage — Essentia 2.1-beta6-dev documentation [Internet]. [cited 2023 May 9]. Available from: http://essentia.upf.edu/

s104: ICCM: Instrument Classification in Carnatic Music

Software development tools, [version 2.0]

Guillem Gauchia, Àlex Herrero, Gerard San Miguel, Roddie Mc Guinness

05/05/2023

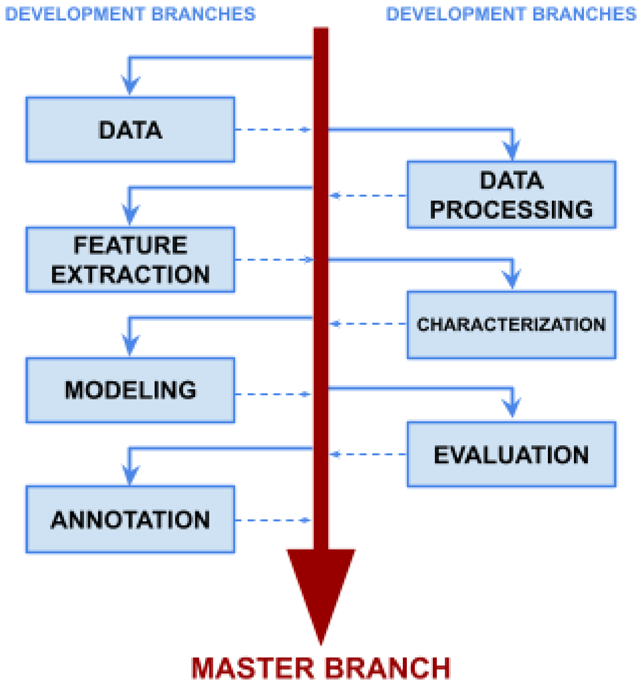
ICCM: Instrument Classification in Carnatic Music is a software development project aimed at creating a machine learning model that can accurately classify musical instruments used in Carnatic music.

Code repository

You can access the Github repository using this link.

Repository workflow

This will be the workflow we'll be using through the whole project:



First, we will create a Master Branch that will contain all the branches that we'll use for each part of the project. Every time one of the development branches is finished or is at a point of functional development, it will be added to the Master Branch, as long as the member responsible for the master branch verifies that it is working correctly and that this latest version does not affect the rest of the project. In order to ensure this, we will use a pull-request system.

The organization of which member/s of the group do each branch will be evenly distributed according to the difficulty of the task, and it can be an individual task or several members working on the same branch. It will be decided which member/s of the group will do each branch during the development.

Repository architecture

Initially, this is the structure of the GitHub repository:

File	Description
README.md	File that provides a description of the fundamental basics on how to use and implement the project software, going step by step over each phase of the whole pipeline.
requirements.txt	File that specifies all the requirements that need to be installed in order to prepare the environment.
/src/	Folder that contains the source code of our whole project.
/model/	Folder that contains all the files and information about the ML model.
main.exe	The program's executable.

The following programs have been created inside the /src/ folder, following the steps of the ML pipeline:

File	Description
Install_dataset.ipynb	Installation of the saraga1.5_carnatic dataset (16.2 GB) using the corresponding functions of the mir-data library. The download might take a long time depending on the connection
Dataset_Creation.py	Creation of the database structure used for the ML model that allows us to detect the instruments that sound during each given time interval.
Feature_Extraction.py	From the data obtained from the creation of the dataset using the program ***Dataset_Creation.py***, the features of each of the audios are obtained to be analyzed. This program extracts individual features in order to later predict the instrument playing on each chunk taking into account some of the following parameters: MFCC, Chromagram, Loudness, Energy, Spectral envelope…
Feature_Characterizact	From the training data many features are obtained and this

ion.py	program allows to determine how each of these features are classified in groups of given instruments.
Predictive_Model.py	After processing the features, a predictive model is developed to analyze different songs in order to classify their instruments and when each one of them sounds.
Results_Evaluation.py	This program compares model results within real results in order to determine the strength of the predictions. Helps to determine how to split the data into train/test samples in order to avoid an overtrained model.
Results_Annotation.py	This program offers a graphical annotation of the tested songs, explaining where each instrument is being played.
main.py	This will be the main program that will execute all the other functions mentioned before, following the pipeline workflow shown in the previous figure.

Software tools

In developing the ICCM model, several software tools and libraries will be used. These include:

- **Python** - Will be the main programming language used to develop the project. Link here to download Python.
- **Jupyter Notebook** - An open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. Link here for an installation guide.
- **Google Colaboratory** - A cloud-based platform that provides free access to a Jupyter Notebook environment with GPU support.
- **Pycharm** - Could be used for the implementation of specific scripts and programs to then implement at the jupyter or Google Colab project. Link here to download Pycharm
- **Scikit-learn** - A machine learning library for Python that provides simple and efficient tools for data mining and data analysis. Link here for an installation guide.
- **Mirdata** - A library common to load Music Information Retrieval (MIR) datasets. Link here to the original repository
- **Essentia** - An open-source library and tools for audio and music analysis, description and synthesis. This will be useful specially for Feature Extraction. Link here for an installation guide.

- **Librosa** - A python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems. This will also prove to be useful for Feature Extraction, since it has some algorithms that Essentia doesn't have, in case we decide to use them. Link here for an installation guide.
- **Matplotlib** - a plotting and visualization python library that provides a high-level interface for creating and customizing various types of plots, charts, and visualizations. Link here for an installation guide.
- **Soundfile** - a Python package that provides functionalities for reading and writing audio files. It allows developers to work with various audio formats, such as WAV and FLAC, providing a straightforward and efficient way to manipulate and process audio data. This will be useful for the Dataset creation and Feature Extraction. Link here for an installation guide.
- **Ipython** - a python package that allows you to display various types of content, including text, images, audio, video, HTML, and more. Link here for an installation guide.
- **Pandas** - a python library designed for data manipulation and analysis. It provides data structures and functions that make it easy to work with structured data, such as tabular or time-series data. Link here for an installation guide.

This list could be expanded to include future software tools used in the implementation of the project.

Collaborative coding strategy

The project utilizes an effective and well-structured collaborative coding technique to maintain a high degree of teamwork. The use of branches, issues, pull requests, debates, and well defined team duties are just a few of the activities that fall under this methodology.

By using this approach, each team member takes responsibility for a particular area of the project, encouraging a sense of accountability and guaranteeing that each area receives the appropriate attention. Any code adjustments are carried out on different branches to avoid conflicts with the main codebase. This permits autonomous work without jeopardizing the stability of the primary code.

Pull requests are the technique for integrating code changes into the main source, facilitating the smooth integration of changes. This strategy encourages teamwork and permits extensive code reviews, ensuring that all changes are approved before being implemented and that they fulfill the project's quality criteria.

S104: ICCM: Instrument Classification in Carnatic Music

Software requirements specification, [version 1.0]

Guillem Gauchia, Àlex Herrero, Gerard San Miguel, Roddie Mc Guinness
11/05/2023

Purpose

Instrument Classification in Carnatic Music (ICCM) is a project developed for the Music Technology Lab optative course, at Universitat Pompeu Fabra.

System overview

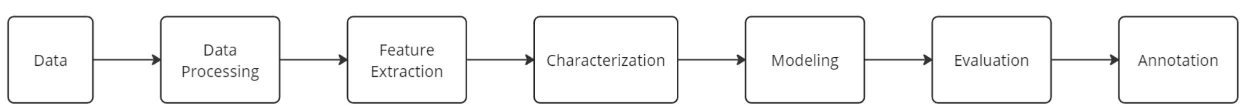
It aims to create a tool for musicologists, musicians and enthusiasts alike to identify and visualize the presence of the main instruments found in Carnatic music tradition (human voice, violin, tanbura and mridangam) in a piece of the user's choice.

This will be achieved through the training of a ML predictive model that will be able to classify each instrument, ultimately creating a graphical representation of the in and outs of each instrument through the length of a performance.

Overall description

Machine Learning pipeline

For the development of this project, we will be using an usual ML pipeline consisting of the following steps:



- **Data and Data processing**: creation of the dataset and cleaning of the samples through audio processing procedures.

- **Feature extraction** and **Characterization**: extraction of the features we will be using to characterize each instrument.
- **Modeling**: construction of a prediction model based on those characterizations.
- **Evaluation**: quantification and evaluation of the performance of our model.
- **Annotation**: development of a visualization process that will be able to identify in a timeline the separate instruments, using a mixed audio file as an input for our model.

Data and Data Processing

The data and data processing phases consists of the following steps:

1. **Dataset creation:**
    a. Access to the saraga1.5_carnatic dataset using the mirdata library.
    b. Retrieval of metadata information and paths of the mixed audio files and the individual instrument files for each performance, using mirdata functionalities.
    c. Loading of the audio files using the librosa library, converting them into arrays of data to work with.
2. **Audio Processing:**
    a. Reduction of vocal bleeding using the spleeter library
3. **Tagging audio:**
    a. Identifying silent regions in the isolated instrument tracks using librosa's split function.
    b. Creation of an array for each instrument telling whether it's currently silent/non-silent
4. **Extracting samples:**
    a. Division of the mixed audio data into small chunks using numpy array indexing.
    b. Determine, for each chunk, if every instrument is playing or note (using the previous silent/non-silent arrays)
    c. Indexing and saving each data chunk into an audio file using the soundfile library.
    d. Creation of a metadata dataframe using the pandas library (including relevant track information, the unique chunk index and a column indication whether or not it includes each instrument)

As we progress through the project, we will be adding the detailed process of each phase.

Libraries and other dependencies

In order to correctly prepare the environment for the project's software, these are the versions of the libraries and dependencies that are needed for the execution:

Libraries	Version	Usage
ipython	8.13.2	Audio playing inside a Jupyter Notebook. Comes in handy for the dataset creation phase.
librosa	0.8.1	A very helpful library for our project. Used in the data processing (audio file into amplitude array, identifying silent regions in audio, etc.) and feature extraction (MFCC, RMS, among other algorithms) phases.
matplotlib	3.7.1	2D plotting of audio files. Useful for the data and data processing stages.
mirdata	0.3.7	Access to the Saraga Carnatic dataset.
numpy	1.22.4	Universally used library. Used in the data phase to split mixed audio into small chunks, among other less relevant applications.
pandas	1.5.3	Used to create all the required dataframes of our project: (metadata.csv, features.csv, etc.)
scikit-learn	1.2.2	Useful for data analysis in the evaluation stage.
soundfile	0.12.1	Used to save the audio array chunks into audio files (sample extraction for the creation of the dataset).
spleeter	2.3.2	Isolation of vocal tracks from background instruments (data processing).

The versions mentioned are the ones we've used, and in some cases, it may not be a problem to use different versions than those provided (for instance, other versions of Python, such as 3.11.3, work just fine), but it's advisable to stick to the displayed versions.

UI Design

Although we're not completely rejecting the possibility of creating an UI for our tool, we're striving to get the tool working on a command prompt environment. Regardless, we made an early interactive prototype for this hypothetical UI. An alternative could be the use of a CLI (command-line interface).

Constraints

- The modeling phase could potentially take too long, so we will be careful about wasting our time and the amount of features taken into consideration for the processing of the model, among other factors.
- As stated above, due to our limited time in the course, we might not be able to develop a working UI for our tool.

Specific requirements

Functional requirements

- **Annotation Visualization:** The system should provide a graphical representation that identifies and visualizes the presence of each instrument in a timeline format.
- **Clear Instructions**: in a text-based environment it's very important to concisely guide the user through our program.

If we end up creating an UI, the amount of functionality requirements could grow:

- **Audio player**: it would be ideal to have an integrated audio player that will let the user compare the audio input with the generated plot.

Hardware and software requirements

- **Programming language**: Python (v3.10.8)
- **Operating system**: The code has been tested on Windows 10 and macOS Ventura machines, but compatibility with other operating systems should be verified.
- **Storage amount**: At the very least, the total database size (16.2 GB) + the training data chunk duplicates (~40-60% of the total size) + the required installations (<5% of total size).

Non-functional requirements

- **Accuracy:** The classification model should achieve a high level of accuracy in identifying and classifying each instrument (human voice, violin, tanbura, and mridangam).

- **Performance**: The system should be capable of processing and classifying audio data in real-time or within an acceptable timeframe.
- **Compatibility**: The system should be compatible with commonly used audio file formats (for example, WAV or MP3) to accommodate a wide range of user inputs.
- **Error-handling:** The system should handle unexpected inputs or errors gracefully, providing informative error messages and preventing crashes or data corruption.

S104: ICCM: Instrument Classification in Carnatic Music

Ethical Considerations, [version 2.0]

Guillem Gauchia, Àlex Herrero, Gerard San Miguel, Roddie Mc Guinness
11/05/2023

Introduction

In this paper, we will address ethical considerations related to our project. With the advent of artificial intelligence and its growing influence in various fields, it is critical to recognise and address the ethical challenges that arise in the development and implementation of these systems.

In this introductory section, we will highlight four key ethics issues that are relevant to our project: algorithmic bias, social impact, open science, and gender bias. These issues provide an essential framework for assessing and considering the ethical implications of our work.

By addressing these key ethical issues in our project, we can ensure that our work is conducted responsibly and benefits the community of musicians and listeners without causing harm.

Ethics of AI

To take into account the ethics carried out by the artificial intelligence in our project, the data used to model our project has been considered, ensuring that the data used for training and evaluation purposes has been obtained and used in a way that respects the privacy and consent of the performers in the Saraga dataset.

After the data had been prepared and classified, the instrument classification model was developed in a transparent and explainable manner. The AI system's decisions were designed to be understandable and interpretable, providing information about the reasoning behind the classification results. Interested parties can refer to our development reports and open-source GitHub repository for more details on the model's inner workings.

Furthermore, the development team took the responsibility of monitoring the system's performance and addressing any biases or undesirable outcomes. This ensured that any issues were promptly addressed and rectified.

Algorithmic bias

Taking into account the possible problems of our project having algorithmic bias, we have analysed what could be the possible errors that lead to it. In our case we have seen that the only option to generate this algorithmic bias is by using an uneven dataset.

So that the samples treated in the Machine Learning model, for the classification of instruments in Carnatic music, are independent and identically distributed, we have created a program that, given the data of the database which includes in each song the different audio tracks where each of the instruments (voice, violin and mridangam) sound in isolation, this program allows us to classify in small intervals of time which instrument or instruments as a whole sound in each chunk. In this way we have classified in 8 possible combinations which are:

1. None of the instruments
2. Vocal
3. Violin
4. Mridangam
5. Vocal and Violin
6. Vocal and Mridangam
7. Violin and Mridangam
8. Vocal, Violin and Mridangam

With this classification we can obtain the same number of each of the possibilities and thus be able to generate a balanced model for each instrument.

Social and Cultural Limitations

Firstly, it is important to acknowledge that not all instruments of the Carnatic music tradition are included in the analysis due to limitations in the available dataset. The ML model only focuses on identifying the human voice, violin, tanbura, and mridangam. This limited scope may inadvertently overlook the contributions of other significant instruments in Carnatic music.

It is also crucial to recognize that we are not carnatic musicians ourselves. Despite some of us having a musical background, we lack first-hand knowledge and understanding of this music tradition and culture, since we are all based in Europe and haven't had much contact with Indian traditions. In this sense, the input of a Carnatic music expert would have been ideal.

Despite all these limitations to take into consideration, our idea with this project is to provide a useful tool for those who want to work on the study and classification of the most significant instruments in songs of the Carnatic genre. On the other hand, the realization of our project can bring visibility to the genre of Carnatic music and thus make it known in a more globalized way, so that people who do not know what is Carnatic music have the possibility not only to know a music coming from a different culture (South India) for certain people, but also to know in a better way the instruments that have an important role in the songs of this genre.

Open Science

Our project is intended to be completely Open-Source since mainly the implementations of our code are based on open source libraries and the data used are also free to use for projects such as in our case. In addition, being a university project, as a group we do not expect to obtain any financial gain and simply to progress in our academic course as well as gain knowledge in the subjects of Machine Learning and its adaptation in music, specifically in Carnatic music.

Gender Bias

We also have to take into account that when selecting the saraga_1.5 dataset we have not given importance to the gender of the performers. The dataset itself may have a disproportionate representation of gender, which could be significant for the characterization of the human voice (since there is a presence of both female and male singers in our dataset). We have trained a model on the human voice as a whole, trying to have a representative pool of both genders, but we are aware of the limitations this might have (in comparison to training a model on female and male singers separately).

S104: Instrumentation Classification in Carnatic Music

Evaluation

Guillem Gauchia, Àlex Herrero, Gerard San Miguel, Roddie Mc Guinness

20/06/2023

Introduction

This project is based on the creation of a tool that allows, by means of a Carnatic music song, to identify at which moment of a certain song, at the user's choice, which of the three main musical instruments that make up this musical genre are played. These three instruments are the voice of the artist, the violin and the mridangam which is divided into left and right, however, when both drums are analysed they are treated as one.

To obtain results of the model created that allows us to make the prediction of the instruments, we have created a dataset where dozens of songs have been analysed, in this way we have generated the samples that are based on chunks of a size of 0.25s, and for each of these chunks, we have the information of which instruments sound and the audio track that corresponds to it. From the creation of the dataset, the next step is based on obtaining the features that allow us to classify to which class of the following combinations of instruments it belongs:

0. None

1: Vocal

2. Violin

3. Mridangam

4. Vocal + Violin

5. Vocal + Mridangam

6. Violin + Mridangam

7. Vocal + Violin + Mridangam

Obtaining the features allows us to later create the prediction model that will be used to determine which instruments are played at which point in the song.

Model

The data for training and testing the model will be distributed following approximately 75% for training and 25% for testing, leaving a hold-out that allows us to obtain real results and determine the effectiveness of the model.

Mainly, we doubted which model to choose, but finally we have used "Gradient Boosting Classifier", we have used this model because in comparison with our other option "Histogram Gradient Boosting Classifier" the first one works better when the number of samples is less than 10.000 and in our case we work with around 8.000 samples that are equally classified among the eight possibilities mentioned above. In order to obtain a better result in the creation of the predictor model, we have decided to split the model into the three instruments (Voice, Violin and Mridangam) so that we can only classify if one of the instruments appears individually.
In this way the process of creating the model is based on the following steps:

1. We divide the features into training, test and holdout data in a random way to try to obtain independent and identically distributed data, in addition to obtaining the target of each sample that determines the class to predict.
2. We create a function to select the same number of samples from each of the 8 possibilities, for example if we have 8,000 samples we will have 1,000 samples for each possibility.
3. Before creating the model we start a Cross-Evaluation using Grid Search that allows us to determine which are the best tuning parameters of the model by testing all possible combinations of parameters (learning_rate, max_depth, n_estimators).
4. Once we have defined the optimal parameters for the three models, we create the model and check the results obtained.
5. We make predictions and observe how effective the model is.

Metrics & Hypothesis

Turning to the practice with the real data, we can determine the following conclusions based on the results obtained during the process of creating the model and selecting its parameters.

Firstly, when we try to make the distribution of samples for each of the eight possible combinations of instruments, we can observe that due to the lack of samples of certain types, we cannot obtain a balanced distribution with a sufficient and minimum amount of samples to train the model. The distribution obtained can be seen in the following screenshot:

```
Sample Distribution:
[356, 839, 1000, 28, 1000, 107, 314, 1000]
```

This is mainly due to the fact that not all instruments sound at the same frequency as each other, and what we mainly thought would be a good amount of samples to work on (+20.000), this amount does not really supply for those instruments, such as mridangam, that sound for less time in the songs. That's why we observe in the example that only 28 samples have been obtained from chunks in which only mridangam is played or for example 107 samples of voice and mridangam.

Following this we wanted to analyse the distribution of ones and zeros for each of the targets of each instrument, where the ideal would be to obtain a quantity of equivalent ones and zeros, the result has not been this, but for the cases of voice and violin, we can observe that there are more ones than zeros, while for mridangam the situation is the opposite, being much more disparate in the second and third case:

```
Percentage of ones in voice:
0.6343669250645995
Percentage of ones in voice at training set:
0.6635
```

```
Percentage of ones in violin at training:
0.7136089577950043
Percentage of ones in violin at training set:
0.741
```

```
Percentage of ones in mridangam:
0.312015503875969
Percentage of ones in mridangam at training set:
0.3385
```

In any case, we wanted to analyse the best parameters for each of the three models on the basis of these data, even if they are biased, and these are the results obtained by Grid Search, comparing them with the parameter values most commonly used in this type of models:

Voice → {'learning_rate': 0.01, 'max_depth':3, 'n_estimators':250}

```
Best parameters are: {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 250}


0.662 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 5}
0.662 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 50}
0.662 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 250}
0.662 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 5}
0.662 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 50}
0.663 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 250}
0.662 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 5}
0.662 + or -0.002 for the {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 50}
0.657 + or -0.006 for the {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 250}
0.662 + or -0.001 for the {'learning_rate': 0.1, 'max_depth': 1, 'n_estimators': 5}
0.662 + or -0.001 for the {'learning_rate': 0.1, 'max_depth': 1, 'n_estimators': 50}
0.661 + or -0.005 for the {'learning_rate': 0.1, 'max_depth': 1, 'n_estimators': 250}
0.662 + or -0.001 for the {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 5}
0.658 + or -0.004 for the {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}
0.648 + or -0.008 for the {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 250}
0.662 + or -0.004 for the {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 5}
0.64 + or -0.01 for the {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 50}
0.628 + or -0.014 for the {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 250}
0.66 + or -0.004 for the {'learning_rate': 1, 'max_depth': 1, 'n_estimators': 5}
0.65 + or -0.009 for the {'learning_rate': 1, 'max_depth': 1, 'n_estimators': 50}
0.619 + or -0.011 for the {'learning_rate': 1, 'max_depth': 1, 'n_estimators': 250}
0.642 + or -0.011 for the {'learning_rate': 1, 'max_depth': 3, 'n_estimators': 5}
0.586 + or -0.027 for the {'learning_rate': 1, 'max_depth': 3, 'n_estimators': 50}
0.59 + or -0.017 for the {'learning_rate': 1, 'max_depth': 3, 'n_estimators': 250}
0.604 + or -0.007 for the {'learning_rate': 1, 'max_depth': 7, 'n_estimators': 5}
0.592 + or -0.011 for the {'learning_rate': 1, 'max_depth': 7, 'n_estimators': 50}
```

Violin → {'learning_rate': 0.01, 'max_depth':1, 'n_estimators':5}

```
Best parameters are: {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 5}


0.758 + or -0.0 for the {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 5}
0.758 + or -0.0 for the {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 50}
0.758 + or -0.0 for the {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 250}
0.758 + or -0.0 for the {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 5}
0.758 + or -0.0 for the {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 50}
0.755 + or -0.002 for the {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 250}
0.758 + or -0.0 for the {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 5}
0.756 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 50}
0.75 + or -0.004 for the {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 250}
0.758 + or -0.0 for the {'learning_rate': 0.1, 'max_depth': 1, 'n_estimators': 5}
0.758 + or -0.0 for the {'learning_rate': 0.1, 'max_depth': 1, 'n_estimators': 50}
0.754 + or -0.002 for the {'learning_rate': 0.1, 'max_depth': 1, 'n_estimators': 250}
0.756 + or -0.001 for the {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 5}
0.75 + or -0.002 for the {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 50}
0.742 + or -0.005 for the {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 250}
0.755 + or -0.002 for the {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 5}
0.747 + or -0.005 for the {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 50}
0.736 + or -0.007 for the {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 250}
0.749 + or -0.006 for the {'learning_rate': 1, 'max_depth': 1, 'n_estimators': 5}
0.747 + or -0.003 for the {'learning_rate': 1, 'max_depth': 1, 'n_estimators': 50}
0.732 + or -0.009 for the {'learning_rate': 1, 'max_depth': 1, 'n_estimators': 250}
0.737 + or -0.004 for the {'learning_rate': 1, 'max_depth': 3, 'n_estimators': 5}
0.686 + or -0.01 for the {'learning_rate': 1, 'max_depth': 3, 'n_estimators': 50}
0.68 + or -0.008 for the {'learning_rate': 1, 'max_depth': 3, 'n_estimators': 250}
0.69 + or -0.018 for the {'learning_rate': 1, 'max_depth': 7, 'n_estimators': 5}
0.678 + or -0.005 for the {'learning_rate': 1, 'max_depth': 7, 'n_estimators': 50}
0.721 + or -0.006 for the {'learning_rate': 1, 'max_depth': 7, 'n_estimators': 250}
```

Mridangam → {‘learning_rate’: 0.01, ‘max_depth’:1, ‘n_estimators’:500}

```
Best parameters are: {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 500}


0.667 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 5}
0.667 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 50}
0.672 + or -0.003 for the {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 250}
0.672 + or -0.003 for the {'learning_rate': 0.01, 'max_depth': 1, 'n_estimators': 500}
0.667 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 5}
0.669 + or -0.002 for the {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 50}
0.669 + or -0.005 for the {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 250}
0.664 + or -0.008 for the {'learning_rate': 0.01, 'max_depth': 3, 'n_estimators': 500}
0.667 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 5}
0.669 + or -0.002 for the {'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 50}
0.667 + or -0.005 for the {'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 250}
0.658 + or -0.007 for the {'learning_rate': 0.01, 'max_depth': 5, 'n_estimators': 500}
0.667 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 5}
0.669 + or -0.002 for the {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 50}
0.66 + or -0.007 for the {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 250}
0.654 + or -0.007 for the {'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 500}
0.667 + or -0.001 for the {'learning_rate': 0.01, 'max_depth': 9, 'n_estimators': 5}
0.671 + or -0.005 for the {'learning_rate': 0.01, 'max_depth': 9, 'n_estimators': 50}
0.664 + or -0.008 for the {'learning_rate': 0.01, 'max_depth': 9, 'n_estimators': 250}
0.655 + or -0.008 for the {'learning_rate': 0.01, 'max_depth': 9, 'n_estimators': 500}
```

Once the best parameters were obtained, we assigned them to each model and obtained the following prediction accuracy values:

Score for voice model: 0.4518633540372671	Score for violin model: 0.5434782608695652	Score for mridangam model: 0.84472049689441

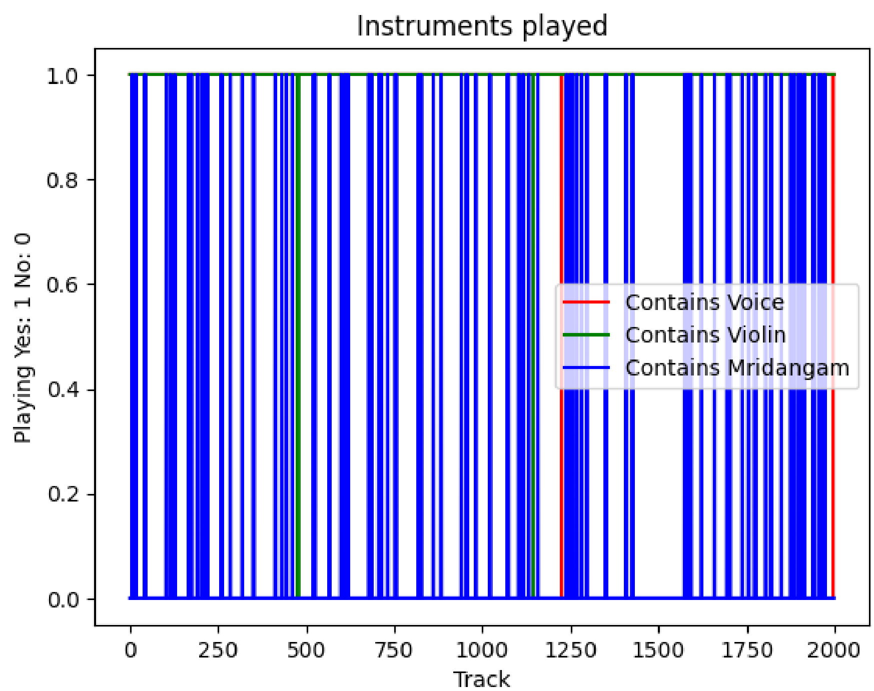
Where these results are clearly not good, and analysing why mridangam has a value that could be considered decent, we see that the model always predicts that there is no mridangam (class 0) and as in the vast majority of samples this instrument does not appear, then it ends up predicting "correctly" because it tends to predict 0 for most samples, this happens for the other two instruments but in reverse, we can observe this behaviour in the following screenshots:

```
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
Prediction that contains voice: 1
```

```
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
Prediction that contains violin: 1
```

```
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
Prediction that contains mridangam: 0
```

Finally, although the results are poor and difficult to work with to obtain a system that allows us to visualise the classification of instruments, this would be a sketch of how the data could be interpreted, in this case with a real case using the current models, the graph that represents in a song how it is indicated when an instrument is playing and when it is not, by means of the prediction of each model.



Conclusions

Once we have seen the results obtained from our model, we can determine that they have been much more negative than we initially expected, being the model very unreliable for classifying instruments. First of all, we believe that one of the possible causes of the model giving bad results is that the dataset we have extracted is not significant enough, perhaps at the moment of detecting silences or the amount of samples obtained has not been enough to create a machine learning model, also as we have been able to observe the number of samples obtained is not evenly distributed.

Another possible reason for the poor results of the model may be that the model selected was not suitable for the context of the data we were working with, or that the way of structuring the models and deciding to create three models for each instrument instead of one model that predicts multiple classes may have been a bad decision when carrying out the project.

With all this we can conclude that the model obtained has not proved to be reliable in the prediction and classification of instruments in Carnatic music songs is not very reliable.

S104: Instrumentation Categorisation and Characterisation in Carnatic Music

Individual weekly update

[Team member]

[17/04 to 27/04]

Work done this week

- individual research about Carnatic Music
- research on the use of github to remember how to use it to be ready for the project
- install all the applications required for the project, such as pycharm

Work planned for next week

- finish the first version of the project plan before friday
- start the project on github
- start the database of carnatic music

[28/04 to 07/05]

Work done this week

- Finish the first version of the Project Plan with Alex and Roddie.
    - More specifically, I worked in the introduction and team responsibilities sections with Alex.
- We decide (all the members of the group) how to divide the different tasks of the project (some of them are not decided yet), depending on the interest and ability of every member.
- Download the dataset that Alex and I will use to select the audios that we consider more important or more useful to the aim of the project.

Work planned for next week

- Finish the dataset with Alex.
- Meet with the teacher and the rest of the group in class.
- Change the Project Plan if it's necessary depending on the teacher's recommendations.

[08/05 to 12/05]

Work done this week

- Work on the dataset functions with Alex (but not finished yet)
- Update the Project Plan

Work planned for next week

- Finish the dataset
- Start with the feature extraction functions
- Finish all the deliverables for this week

[13/05 to 18/05]

Work done this week

- Finish the dataset functions with Alex

Work planned for next week

- Finish the feature extraction functions
- Start with the ia software development

[19/05 to 26/05]

Work done this week

- Finished all the functionalities of the dataset creation with Alex.
- Finish the second version of the Software development tools.

Work planned for next week

- Try to finish the feature extraction code
- Finish all the deliverable documents in process that we have to deliver for the next week

[29/05 to 04/06]

Work done this week

- Fixing the program of the Data Creation to take an amount of samples.

Work planned for next week

- Start working and finish the Ethical Considerations Document.
- Obtain a great amount of samples to work with.

[05/05 to 09/06]

Work done this week

- Finish the ethical consideration document.

Work planned for next week

- Start working on the model.
- Change the dataset.csv that we will use for training the model,

[09/05 to 12/06]

Work done this week

- Working on the model

Work planned for next week

- Finish the model
- Create the slides for the final presentation.

[13/05 to 29/06]

Work done this week

- Finish the model
- Start and finish the slides (on powerpoint or canvas for example) for the final presentation.
- Check all the documents that we have done to update it and combine it in one single document.

S104: Instrumentation Categorisation and Characterisation in Carnatic Music

Individual weekly update

Àlex Herrero Díaz - 240799

[17/04 to 27/04]

Work done this week

- We have met the whole group with the teacher to determine the objectives of the project
- Individually I have searched for information about carnatic music in addition to starting to listen to some songs to understand this musical genre in a better way.
- I have prepared the software environment on my personal laptop to start working on the project
- I have reviewed the contents offered by the teacher in such a way that I have begun to understand the bases of carnatic music as well as relevant information for the programming of the project
- Due to the sudden change of group we have met online with the teacher and the new members to know how to start organizing the project

Work planned for next week

- Get to know the project group better to understand how to organize the work
- Better understand how to start developing the main idea of the project
- Make the next delivery: Project plan V1

[28/04 to 07/05]

Work done this week

- I have worked on the Project Plan V1.0 where we have established certain aspects such as the responsibilities of the members of the group, the work dynamics and how to orient the work to be carried out for the project.
- We have filled in the software development tools document with the elements that we believe will initially be important in order to establish the software environment for our project.
- As responsible for the first element of the pipeline, the Data, Guillem and I have downloaded and reviewed the contents of the mir-dataset database where we will be able to extract pieces of carnatic music useful for the development of the machine learning software for the project.

Work planned for next week

- Work with the data to extract useful information to work with from the dataset.
- Informing us on how to work with the group for the realisation of the next components of the project.
- Update the contents of the submitted documents if necessary once we have discussed the work plan with the tutor.

[08/05 to 14/05]

Work done this week

- Started working on the dataset creation with Guillem
- Updated the Project Plan
- Uploaded the Software Requirements Tools document
- Started to get in touch with feature extraction foundations.

Work planned for next week

- Finish the dataset creation part
- Finish pending documents that have a deadline at the next week

[15/05 to 21/05]

Work done this week

- Finished most of the functionalities of the dataset creation with Guillem.

Work planned for next week

- Start working in the following steps of the pipeline mentioned at the Project Plan.
- Work on polishing the programming code and try to abstract most parts of it.
- Finish pending documents that have a deadline next week.

[22/05 to 28/05]

Work done this week

- Finished the dataset creation Notebook.
- Finished the Software Requirements Tools V1 document

Work planned for next week

- Start working on Feature Extraction Notebook.
- Finish and update pending documents that have a deadline next week.

[29/05 to 04/06]

Work done this week

- Fixing and testing the program to take a huge amount of samples of the Data Creation

Work planned for next week

- Start working on Ethical Considerations Document.
- Obtain a great amount of samples to work with.

[05/05 to 11/06]

Work done this week

- Obtained a huge amount of samples (+ 30.000) in order to obtain features from it.

Work planned for next week

- Start working on Modelling.
- Finish and update pending documents that have a deadline next week.

[09/05 to 12/06]

Work done this week

- Working on the model and trying to test with more data

Work planned for next week

- Finish the model trying to obtain better results
- Start creating the slides for the final presentation and structure the project files to be delivered.

[13/05 to 29/06]

Work done this week

- Finish the model and the presentation
- Prepare my part of the presentation (modeling part)
- Check all the documents that we have done to update it and combine it in one single document.
- Give better format to the notebooks code.

S104: Instrumentation Classification in Carnatic Music

Individual weekly update

Mc Guinness, Roddie

09/06 to 28/06

Work done this week

- Fixed some issues with the features.csv and the extraction process.
- Solved issue with the modeling which was giving 100% accuracy due to an additional feature which accidentally encoded the instrument type in it.
- Prepared my section of the presentation (feature extraction).

31/05 to 09/06

Work done this week

- Produced features.csv file after running and fixing several issues with the Feature Extraction notebook.
- Discussed the modeling stage.

Work planned for next week

- Complete the modeling stage and begin working on the evaluation.
- Submit the evaluation document.

26/05 to 31/05

Work done this week

- Completed a large portion of the Feature Extraction Notebook and finished the State of the Art version 2 document.
- Researched our modeling options some more, we have decided to use Gradient Boosted Trees.

Work planned for next week

- Run all of the Feature Extraction Notebook and produce the features.csv file.
- Work on producing the code for our model.

17/05 to 26/05

Work done this week

- Contributed to the State of the Art version 2 document
- Did some more research on the choice of features and which model we should use.

Work planned for next week

- Complete the Feature Extraction Notebook

12/05 to 17/05

Work done this week

- Helped Alex with some bugs and issues in the Dataset Creation notebook.
- Researched features which would be relevant to extract for our particular case.

Work planned for next week

- State of the Art version 2 document.

08/05 to 12/05

Work done this week

- Completed State of the Art version 1 document.
    - This included reading and researching various papers, topics, and applications that may relate to our project, and summarizing them in a concise form.
- Worked on understanding some of the code from Monday's seminar.

Work planned for next week

- Complete State of the Art version 2 document.
- Work on Data Cleaning and Processing in python notebooks.

29/04 to 07/05

Work done this week

- Discussed with the group how to divide up responsibilities relating to the project.
- Collaborated with the rest of the group to assign tasks to those most suited, in terms of interest, ability, and experience.
- Helped Alex and Guillem with completion of Project Plan v1 deliverable. In particular:
    - I proofread some of the earlier sections,
    - I laid out the risk management section, brainstormed and researched which common and which specific risks might be associated with our project, and investigated potential solutions for some of the issues which we believe may arise.

- Began research and discussion of the State of the Art deliverable, as this document is my responsibility. Completing version 1 of this document tomorrow, May 8th.
- Educated myself on how a state of the art document should look by reading the Wikipedia page

Work planned for next week
- Begin drafting State of the Art v2 document.
- Assist in drafting, editing, modification, and completion of Project Plan v2 document.
- Further planning within the group to figure out how best to proceed.

17/04 to 28/04

Work done this week:
- Met with Thomas after group change to catch up on missed seminar as a result of group change.
- Read additional resources on Carnatic music:
  https://mtl-2023.slack.com/archives/C052G8RJ8BC/p1681714533100269
  And on accessing datasets:
  https://mtl-2023.slack.com/archives/C052G8RJ8BC/p1681719349567099
- Listened to Carnatic music to become more familiar with the tradition, as suggested by Thomas.
- Began to organize the shared drive by creating some basic heading folders, and creating an individual weekly report file for each member.
- Prepared environment on my laptop for beginning to work on the project.
- Additional online meeting with Thomas and the group to begin organizing and understanding our goal before the first Project Planning deadline.
- Engaged in organization of times to meet with the group over the next three weeks.

Work planned for next week
- Decide as a group how best to organize and divide up the work for the project.
- Complete the Project Plan V1 document as soon as possible.
- Brainstorm of project structure and development stages.
- Specify more precisely what the project should accomplish.

S104: Instrumentation Classification in Carnatic Music

Individual weekly update

Gerard San Miguel Navarro

12/06 to 28/06

Work done this week

- Solved a problem with the code of the sample selection for the testing of our model which didn't let the model get representative positive and negative data (presence or absence) for each instrument.
- Iteration with the model to find the best hyperparameters
- Trying to think of the problem with the poor accuracy of our model, after implementing the method to compensate for the lack of equal distribution of positive and negative samples in our dataset.
- Updated the github repo with the latest changes to our project, an improved README.md, license, etc.
- Created most of the slides of the presentation.
- Prepared my part of the presentation.

09/06 to 12/05

Work done this week

- Completed the Ethical Considerations document, adding what Thomas recommended us and improving various parts, specially the AI Ethics, Gender Bias and Social and Cultural ones.
- Completed an initial smaller size model
- Started sketching the final presentation

Work planned for next week

- Improve our model tweaking the various features we've found to be problematical (specially the voice related ones, since we're having a consistent 100% accuracy)
- Develop the full-scale model

31/05 to 09/06

Work done this week

- Investigation on the Gradient Tree Boosting Model (how it works, some characteristics…) and how to implement it through scikit's documentation

Work planned for next week

- Finish all necessary code to be able to create the model (and start with a smaller sample)
- Further investigation of the several hyperparameters related to our chosen model, so we get a better idea of how we could improve it.

26/05 to 31/05

Work done this week

- Started working on the Ethical Considerations Deliverable

Work planned for next week

- Finish the Ethical Considerations deliverable
- Finish Software Requirements Specification V2
- Start working on Characterization
- Decide on which model to use and learn how to implement it

17/05 to 26/05

Work done this week

- Worked on several functions of the Dataset creation document alongside Roddie, since they didn't work properly.
- Finished the Software Requirements Specification document.
- Improved the Software Development Tools document (V2)
- Improved the Project Plan (V2)
- Updated the Github repository
    - Update outdated information (title, description, etc.)
    - Incorporate the structure we planned (src, README.md, etc.)
    - Uploaded requirements.txt

Work planned for next week

- Finish Feature Extraction
- Start working on the Ethical Considerations Deliverable
- Translate the dataset_creation notebook into a runnable .py file
- Improve the Software Requirements Specification (V2)

12/05 to 17/05

Work done this week

- Worked on the Software Requirements Specification deliverable.
- Not much else, I was sick this week.

Work planned for next week

- Finish the Software Requirements Specification document ASAP.

07/05 to 12/05

Work done this week

- Installed the dataset and all the dependencies we needed for finishing the Data phase of our project in my mom's laptop, in order to advance in our sessions with Thomas.
    - Revised our first version of the Project Plan and improved it with suggestions. I also identified some things (such as the calendar) that are slightly outdated, since we've been making some fundamental changes these last days. I'm waiting for this Friday's session with Thomas to get his opinion on the whole document, so we can proceed.
- Started working on the Software Requirements document.
    - I've used both the 101's group version and my notes from the second class of this course (where Xavier explained with detail the different deliverables of the course) as references to elaborate a series of bullet points about what to talk in each section of the document. This way, Friday, after talking with Thomas about our approach, we will be able to swiftly finish the document.
    - I also made a mock-up / interactive prototype of the UI of our final "product". I brainstormed some ideas and functionalities just to have some ground to build from (they're not final, just early ideas), and elaborated a simple design using Figma. I've put a link in the Software Requirements documents to both the Figma project and the prototype.

Work planned for next week

- Finish the three deliverables we have due to this week.
- Finish implementing the functions we were working on this week (the ones that get the data we need from each sound file) to proceed to the next phase of our project.
- Resume my investigation on Feature Extraction.

28/04 to 07/05

Work done this week

- Started investigating Features Extraction:
    - Algorithms I've seen:

- MCFF: I've investigated thoroughly this one in particular, understanding its whole process (FFT, Mel-space, etc…)
    - Chromagram
    - Zero Crossing Rate
    - Spectral Centroid
    - SPectral Rolloff
    - RMS
  - Libraries / packages I've found to implement those features:
    - Essentia. This was mentioned by Xavier.
    - Librosa. This one is very popular and has some features Essentia is missing.
- Checked out and improved some of the other deliveries we had for this week.

Work planned for next week
- Decide who will be in charge of the Github master branch.
- Share what we've been investigating with the rest of the group
- Investigate about which Audio Processing procedures we could use to improve and "clean" our data.
- Continue investigating about feature extraction methods and decide on which features interest us the most
- Start implementing feature extraction tools and playing with the dataset we've already set-up.

17/04 to 28/04

Work done this week:
- Get to know the rest of the group (before the team changes and after the team changes)
- Decide on the type of project.
  - Initially, we decided to discard visualization (what the project initially was about) and go towards melodic/pattern identification.
  - We later decided to change it again, this time focusing on instrument characterization, extraction and classification
- Learn about Carnatic Music (using Wikipedia, this source and the resources Thomas gave us on slack):
  - Most crucial concepts:
    - Svara: note
    - Gamaka: ornamentation
    - Sancara: melodic movements
    - Raga: the "ruleset" about which Svara, Gamaka and Sancara should be used in a determinate performance
- Listen to Carnatic Music (Akkarai Sisters (Violin Duet) / 59th Year Ganesha Festival)
- Decide when to meet weekly during the next 3 weeks with my team, using When2Meet.
- Create the Github repo for the project.

Work planned for next week

- Brainstorm work to do and decide roles/responsibilities of the group
- Find gaps in our knowledge and find what we need to investigate
- Finish the initial Project Planning (we had additional time given by Thomas due to our complications with the group changes)