# SMAI Project

## CLASSICAL PIANO MUSIC GENERATION AND CLASSIFICATION

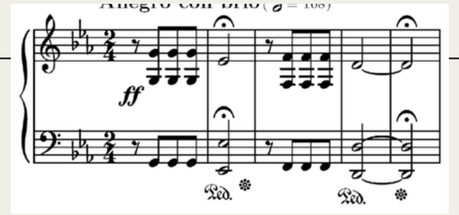
Team 1

- Bhav Beri
- Divij

## OUR AIM

To generate music using Machine Learning models like LSTMs and Encoder-Decoder RNNs and then classify it using a separate music classifier.

# COMPREHENSIVE OVERVIEW

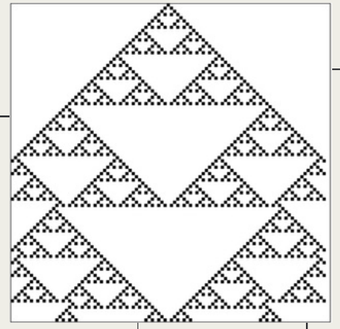


*Understanding MIDI in context of Python*

**Dataset overview**


*Exploration of reference papers*

**Resource review**



*Exploration of non-ML approaches*

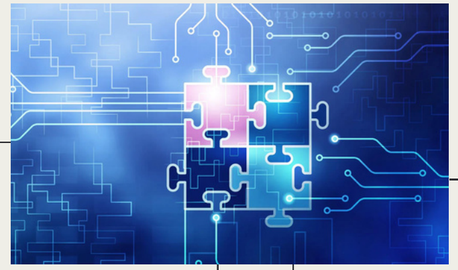
**Cellular Automata**


*Implementation of ML Approaches*

**VNNs, LSTMs, Enc-Dec RNNs**

*Music Classification*

**Feature Extraction**



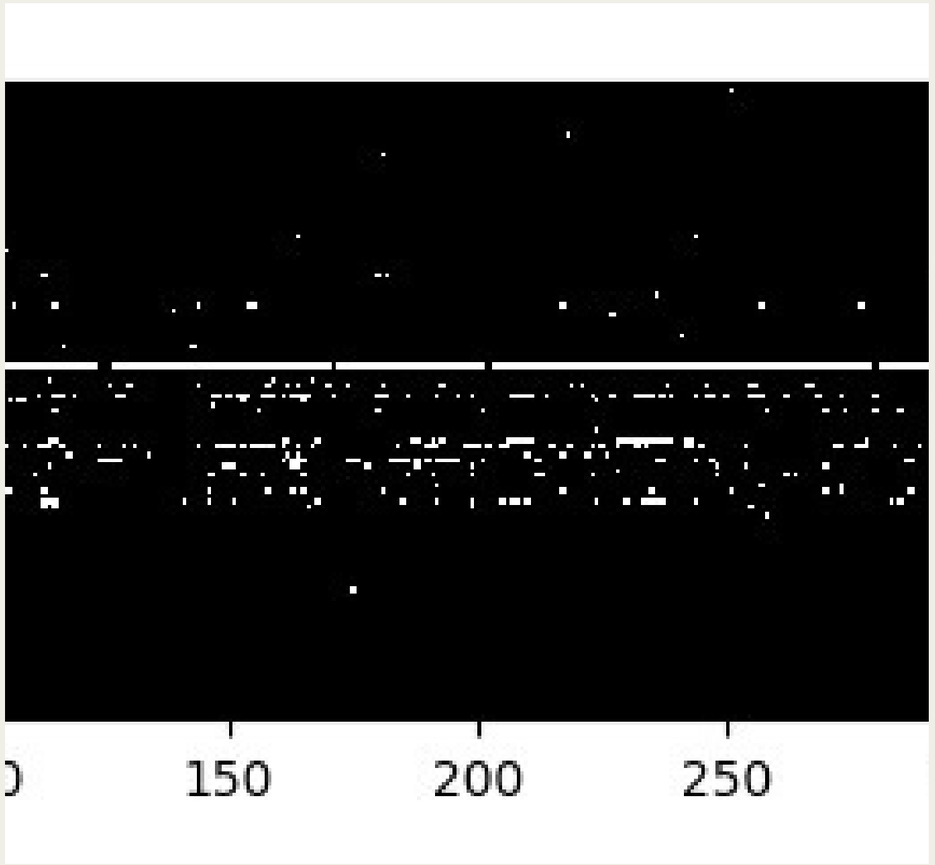
**Combining the systems**

*Challenges faced in Project*

**Project Journey**

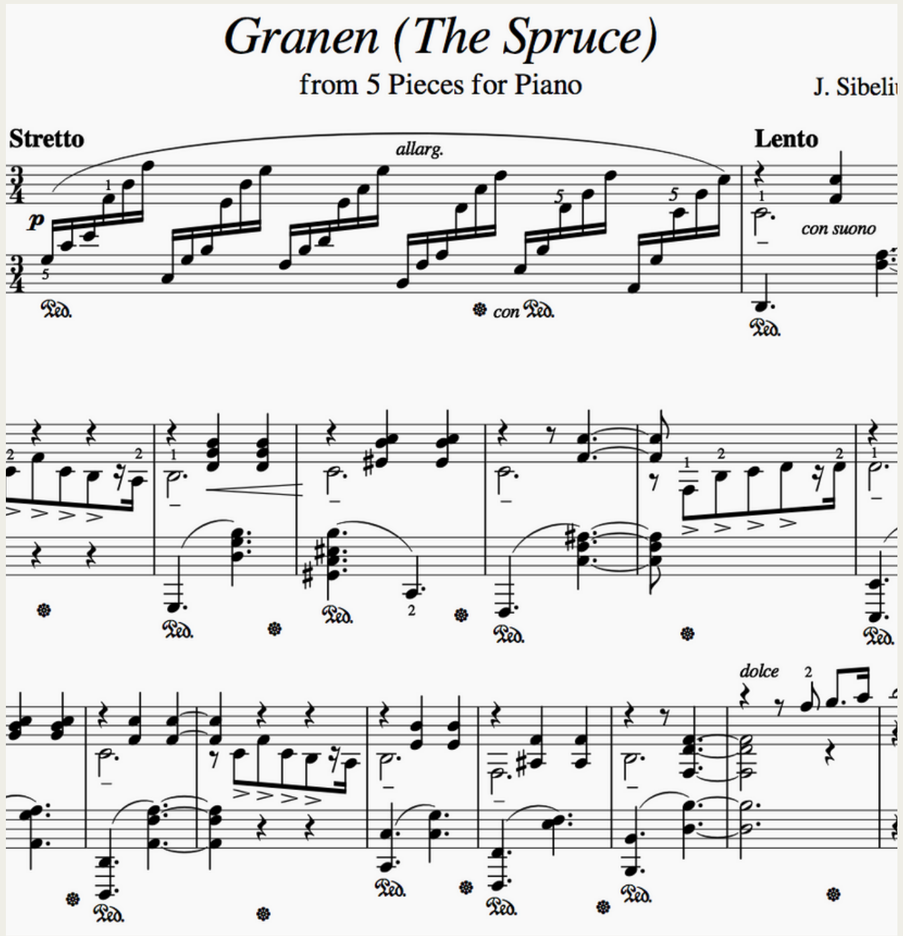
# INTRODUCTION TO MIDI

- Contains musical information such as notes, tempo, etc.
- Rather than audio recordings, they are series of instructions for the music.
- Relation between MIDI and audio file is analogous to SVG and image file.




# MIDI IN PYTHON

- Libraries like music21 and prettyMIDI for interfacing.
- Piano roll: 2-D boolean array
- Columns signify a timestep



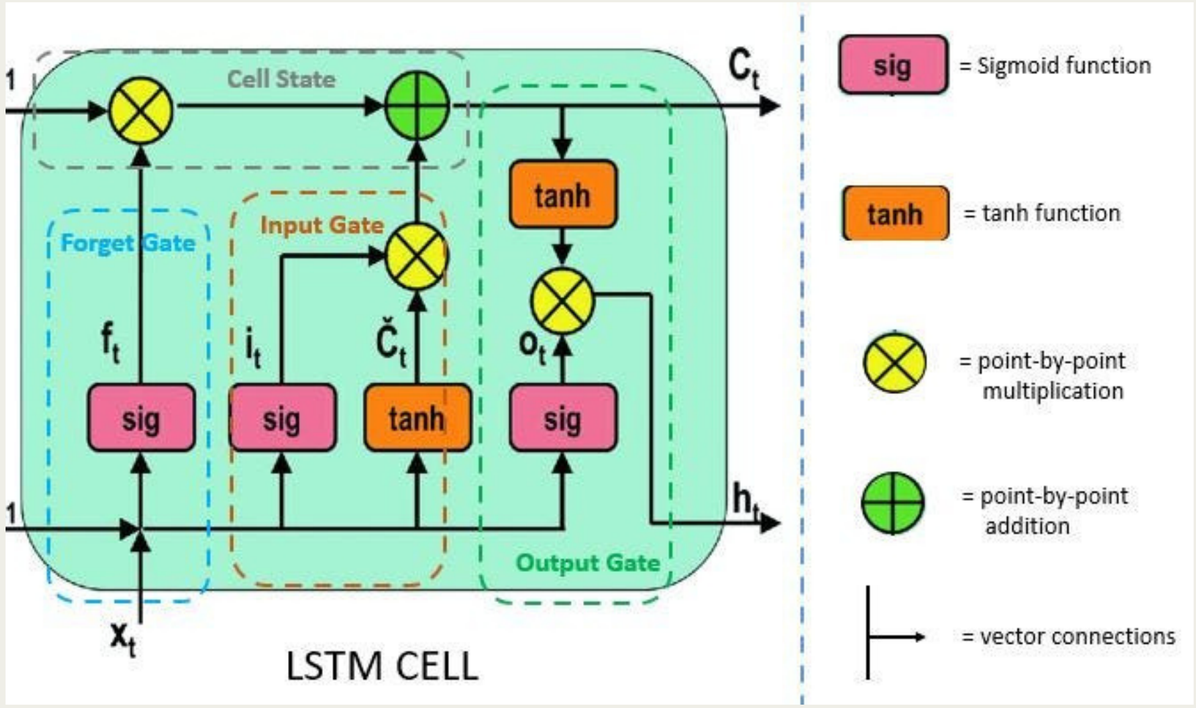

# ISSUES WITH MIDI

- Existing libraries are not documented properly and deprecated.
- These music libraries are only able to interface with some formats of MIDI files.
- MIDI can be written in various ways (multiple instruments, tracks etc.)
- This did not allow us to use various potential data sources.

# REFERENCE PAPERS

- We were given reference papers for Music Generation and Classification.
- The Music Generation paper uses ML models like VNNs, LSTMs and Encoder-Decoder RNNs.
- Their input data are MIDI files for training.
- The Music Classifier paper uses 2-Layer Neural Networks and Deep Softmax Auto-encoders.
- Their input data is a wav file.

# MUSIC GENERATION

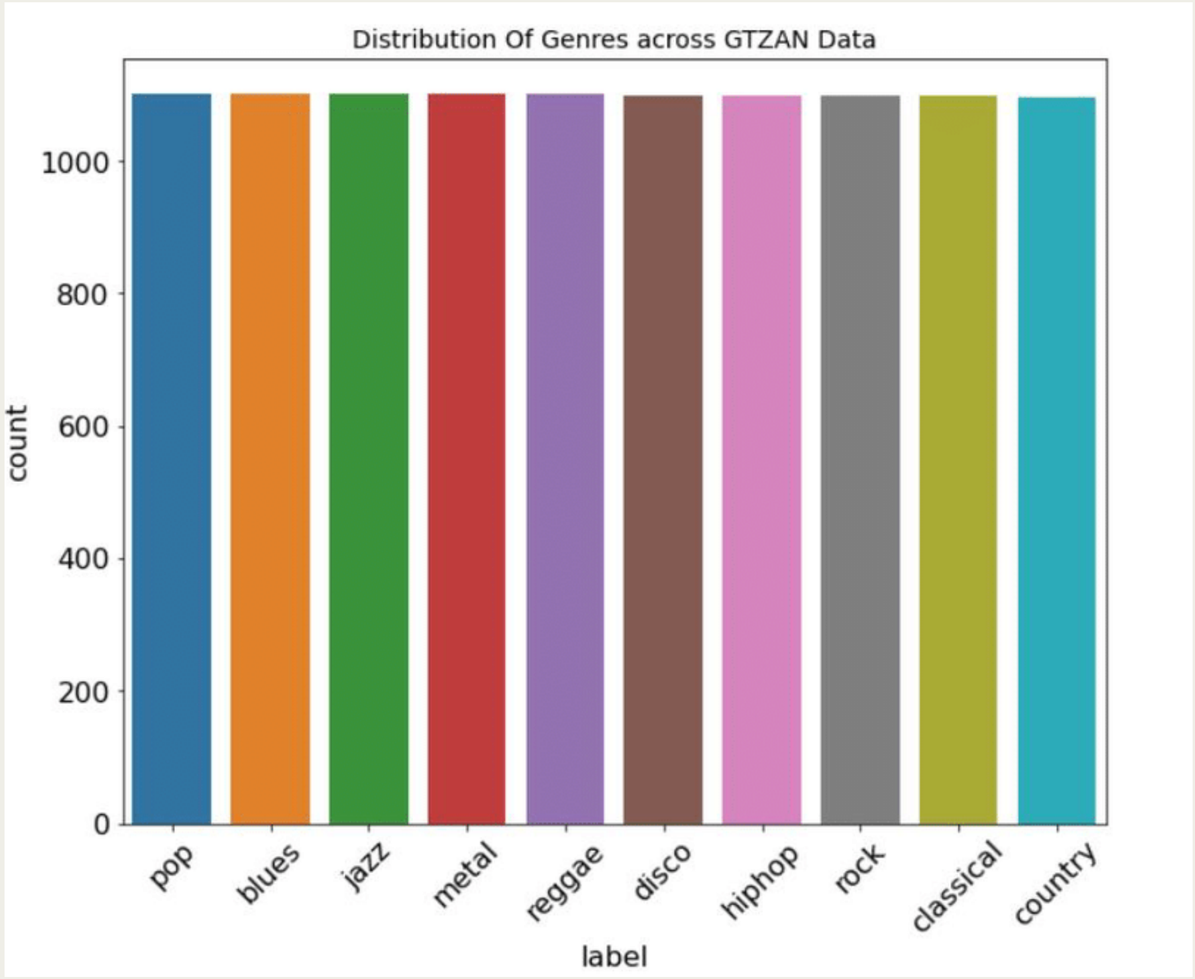



- Their models take a series of keys as input from a MIDI file.
- As an evaluation measure, they find the accuracy of prediction of a note given the previous notes.
- They have used different datasets for different models.

# DIFFERENCES IN OUR WORK

- We have used the same dataset for all models.
- Rationale: Only then can we evaluate the models against each other.
- We have trained the models on individual composers.
- This is done because the style of each composer is different.

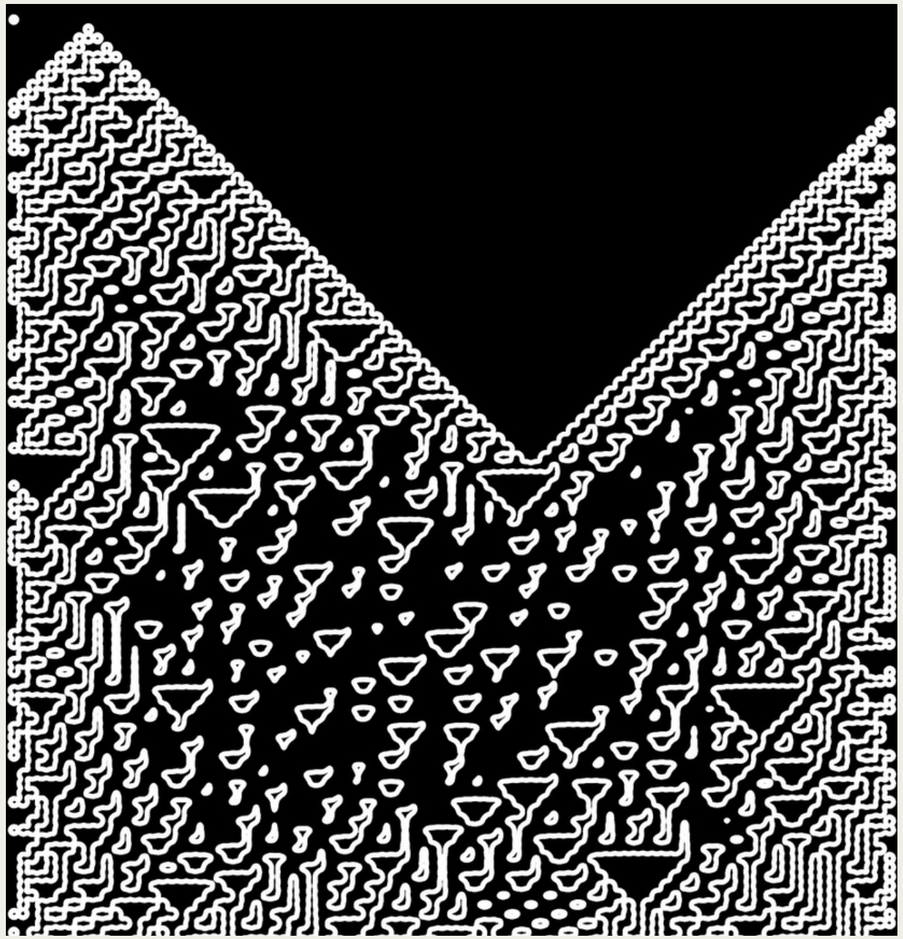
# MUSIC CLASSIFICATION




- Their models take wav file as the input.
- They are extracting the initial 1 second of audio from the wav file for classification.
- They use the standard GTZAN dataset for their model.
- They have done classification on Classical, Jazz, Metal and Pop.

# DIFFERENCES IN OUR WORK

- We have not used the GTZAN dataset because of the difference in our requirements.
- Since our data sources are limited, we have done classification on 3 classes: Jazz, Classical and Pop.
- We have implemented our model differently as their accuracy is low.
- Majority of the times, the initial second is not enough for prediction.
- We have extracted various music features and used Stacking to predict the genre.



# CELLULAR AUTOMATA

- Cellular Automata is a well-known approach for music generation.
- This non-ML approach has shown decent results with some human intervention.
- Wolfram Tones is the most successful project in this area.
- Even simple rules can result in complex music.

# NON-ML LIMITATIONS

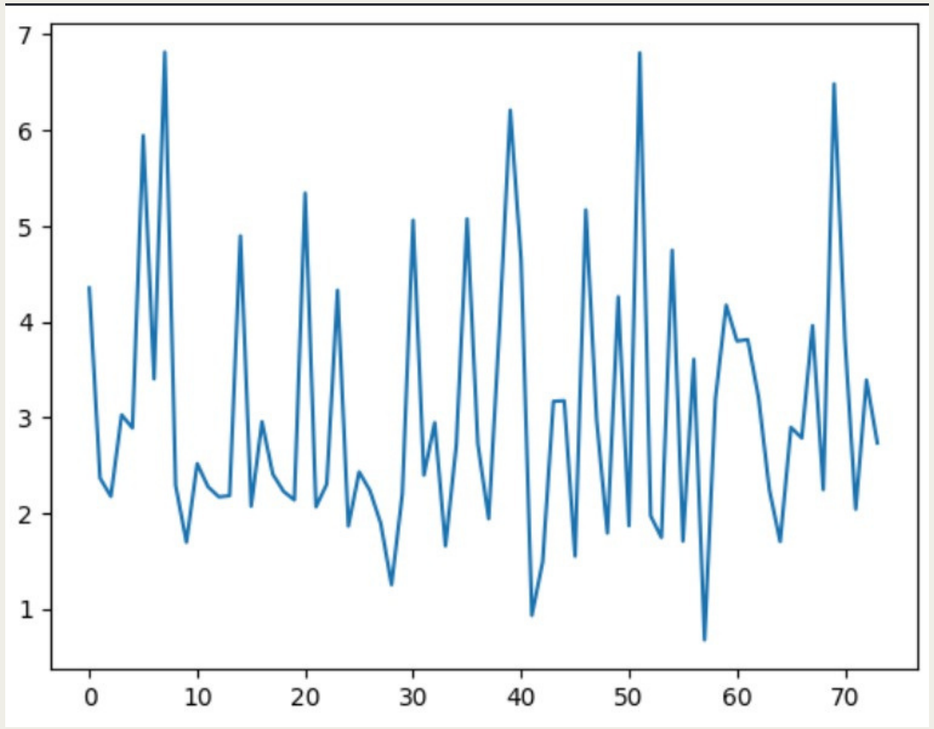
- The music generated is fairly complex, yet clear patterns can be observed overtime.
- This is because of the rule-based nature of cellular automata.
- The music still requires some form of human intervention (defining rules etc.)
- Deterministic nature of music generation results in repetitive outputs.

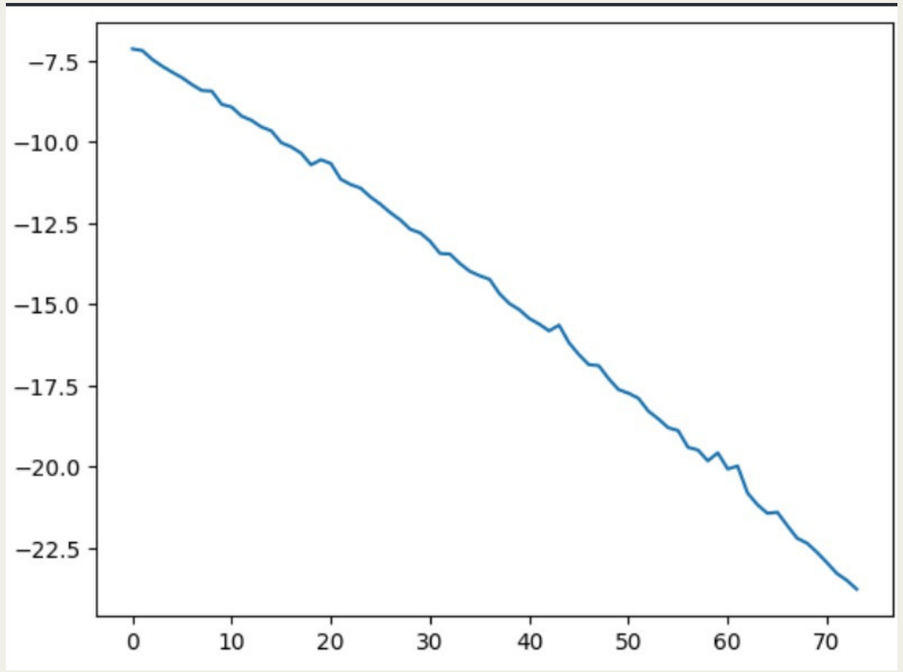
# THE TALE OF 3 MODELS


- Implemented various ML Models for Music Generation.
- Used Triangular Learning Rate for improved generalisation and regularisation effect.
- Implemented Model Checkpointing (saving the best model during training).

# VANILLA NEURAL NETWORK

- Need to give previous keys explicitly since it does not have a memory.
- Using previous 100 keys to predict the next time-step.
- Does not perform as well as LSTMs because of a lack of in-built memory.
- Makes it difficult for learning the patterns since 1s are so scarce throughout the input data.



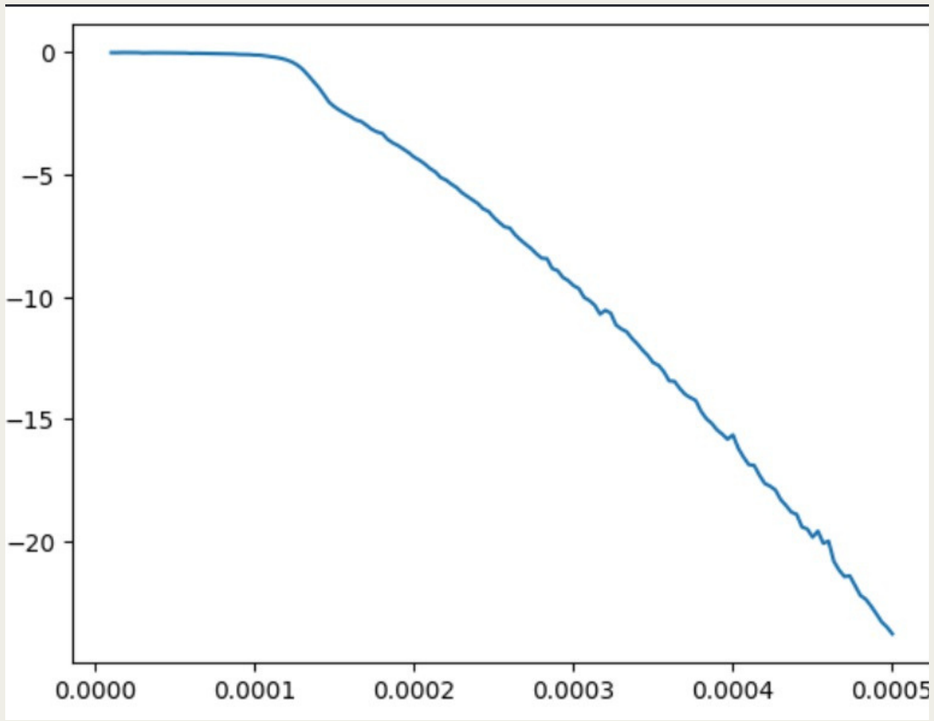


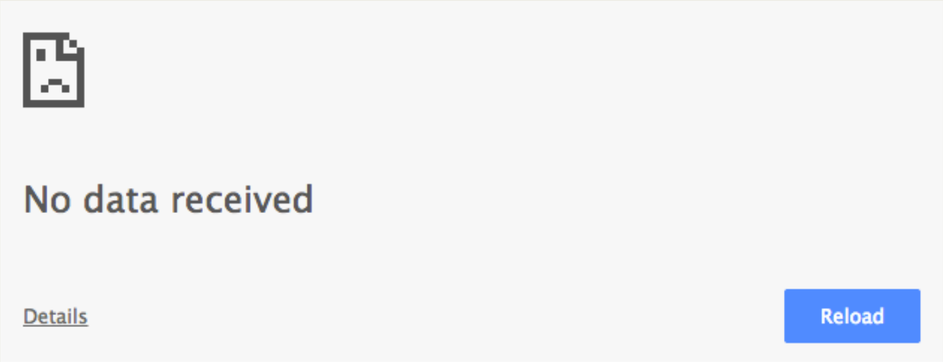

# LONG SHORT TERM MEMORY

- Input are the keys at previous time-step since it has memory capacity.
- Since it can dynamically update it’s memory, the model is able to learn the patterns well (as seen by the high accuracy).
- The music is relatively more complex than other models and coherent with classical music motifs.

# ENCODER DECODER NETWORK

- Similar to LSTM, no need to give all the previous keys because of memory capacity.
- We have used GRUs for the implementation of this network.
- The Encoder-Decoder model has mediocre performance as compared to LSTM.
- Data structure could be a reason as there is minimal changes in each timestep.
- Hence, it may seem as if the notes are repeating.



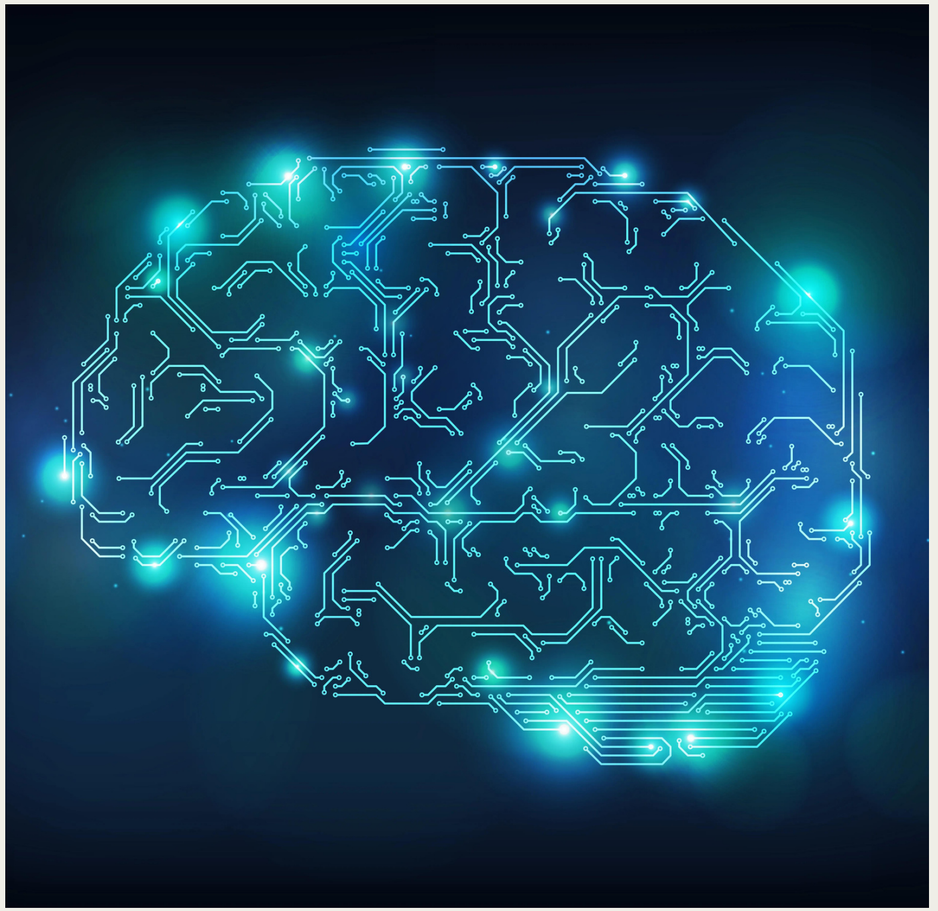


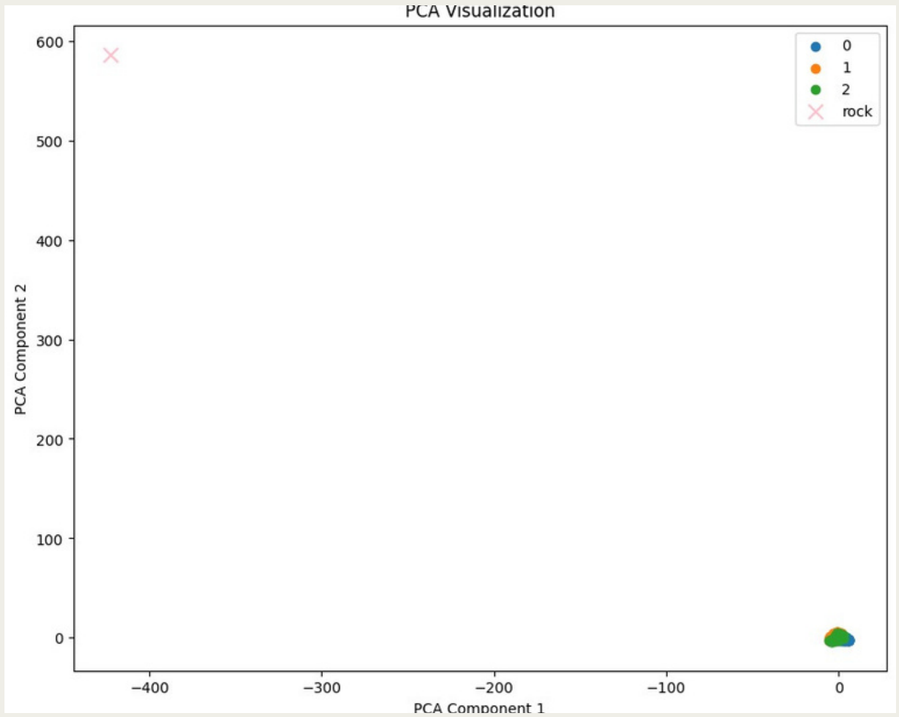

# WHY A DIFFERENT MODEL?

- The model proposed in our classification reference paper uses the first second of audio for predicting the genre of music.
- Unable to replicate their results even after using their own codebase and dataset.
- Unlike their claim, the accuracy of model after using Auto Encoders dropped.

# WHY A DIFFERENT MODEL?

- One possible reason for the drop in accuracy might be because of complexity mismatch.
- The data they are using for training (1 second of audio) is already lacking in the information required for predicting properly (as indicated by the low accuracy).
- Using Auto Encoders can remove further information from the data.





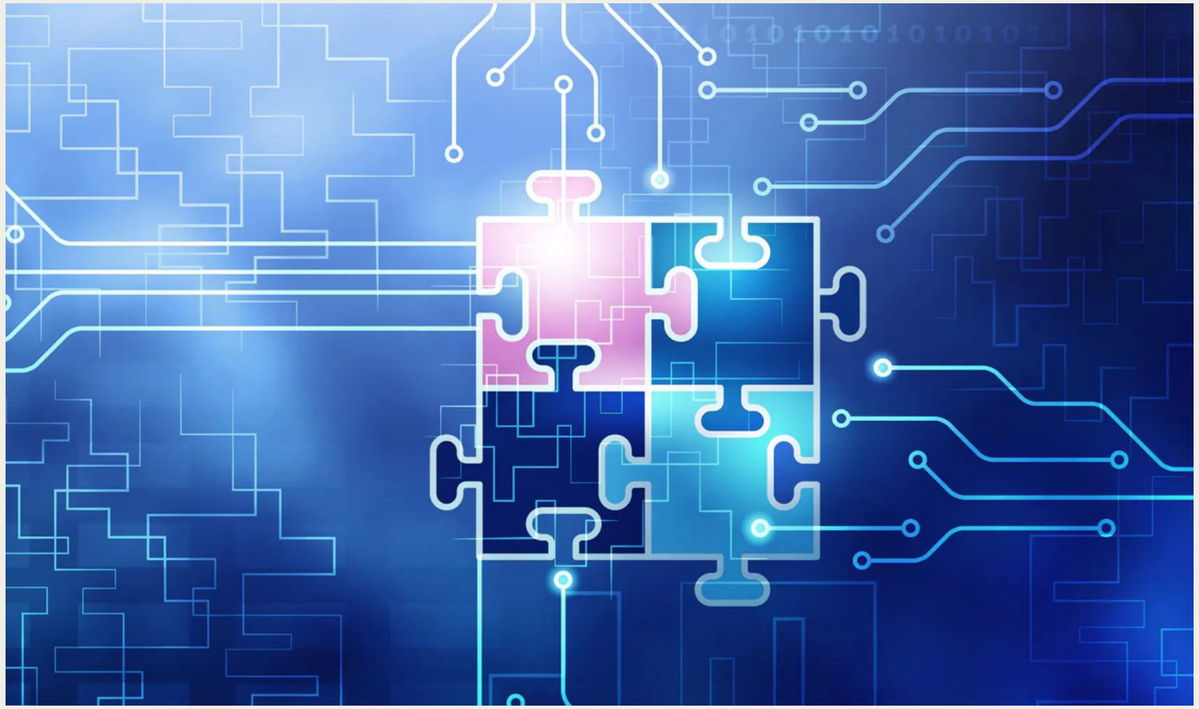

# WHY A DIFFERENT DATASET?

- We have not used the GTZAN dataset for our purpose.
- This is because the music present in GTZAN dataset is vastly different from what we are dealing with.
- Given is a PCA showing the difference between features of their dataset and Beethoven’s musical piece Moonlight Sonata.

# SORTING SOUNDS WITH AI

- Using musical features for predicting the genre of a music piece.
- With the use Stacking, our model reached 81% accuracy on the test-set.
- Our Level-0 models are: KNN, Decision Trees, Random Forest and MLP and the Level-1 Model is Decision Tree.
- We have extracted various features such as tempo, zero-crossing rate and chroma feature.

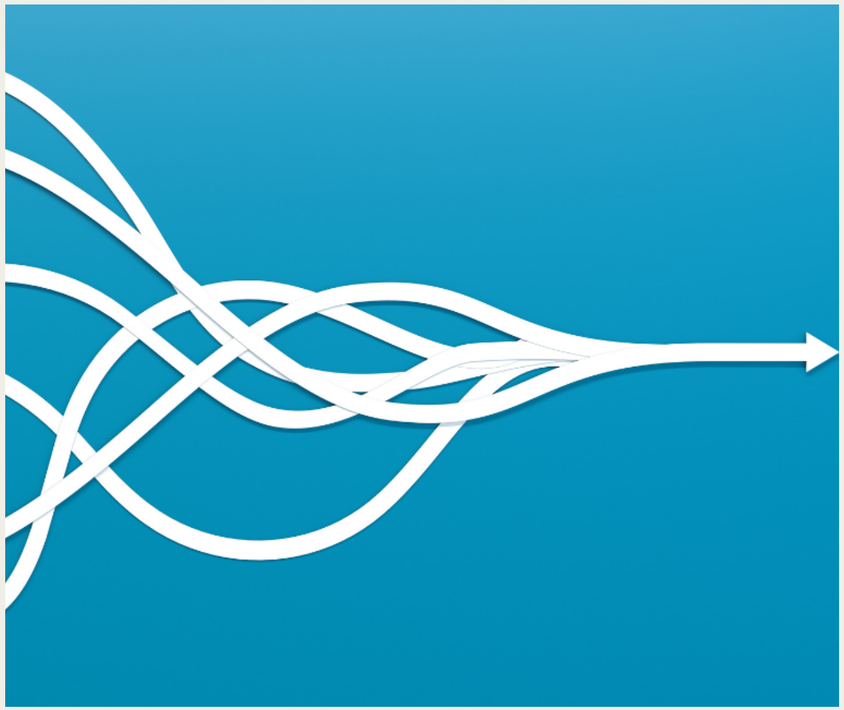
# PUTTING THE PIECES TOGETHER



- The original issue with the project was that the classification model was being used separately from the music generation model.
- Since the datasets were different, we were unable to combine them together for satisfactorily results.
- However, by bridging the gap between the dataset and using a better approach to classification, we have solved this issue.

# COMBINING MODELS

- Used the Music Classification model to predict the generated music from Music Generation model.
- The results were satisfactorily and our generated music was classified as classical as expected.
- We have used Stacking so that various models can contribute to the results.
- This is so that we are able to capture the diverse pattern in music.



# CHALLENGES FACED

## MIDI Files

Dealing with MIDI files is very typical, partly because the libraries for doing so are very old and outdated.
There is little documentation regarding the functionalities and online resources for solving errors, bugs etc. are also scarce.
It took us longer than expected for loading and pre-processing the data because of this reason.

## Reference Papers

The reference papers given to us were not very reliable. The first paper mentioned resources for training generative models but it was not enough. Further, they mentioned about getting 700 MIDI files from a website but we could only find about a 150. The codebase they provided did not have the implementation for Encoder Decoder model so that is not credible either.

The second reference paper mentioned results on the GTZAN dataset for their models. However, we were not able to replicate this using their codebase and dataset.

# CHALLENGES FACED

## Dataset

The datasets used in the two reference papers are completely different. The first paper using classical piano music and reads data from MIDI files while the second paper uses a different dataset and reads from WAV file. The second dataset classifies classical music as something entirely different as compared to the classical piano pieces that we tested upon.

Further, we were unable to find significant number of files for the classical piano music which led to small dataset and sub-optimal training initially. Later on, we spent a lot of time on looking for MIDI files separately from different datasets and extracting the ones that were usable for us. We also used youtube and other such platforms to download music and convert it to MIDI files ourselves. In this way, we were able to overcome this challenge to a greater extent and combine the two models.

# SOTA IN MUSIC GENERATION


- Meta has recently released the Music Gen Stereo model.
- The model is based on Transformers and is able to generate high-quality stereo music (different audio in left and right channels)
- Stereophonic sound, also known as stereo, is a technique used to reproduce sound with depth and direction.

# SOTA IN MUSIC GENERATION


- DeepMind has also released it’s new model called Lyria.
- The AI music generation model is able to generate vocals, lyrics, and background tracks mimicking the style of popular artists.
- The model is based on an auto-regressive neural network architecture and is available experimentally right now.

# Thank you!

**WE HOPE YOU LIKED OUR PROJECT.**

Bhav Beri
Divij