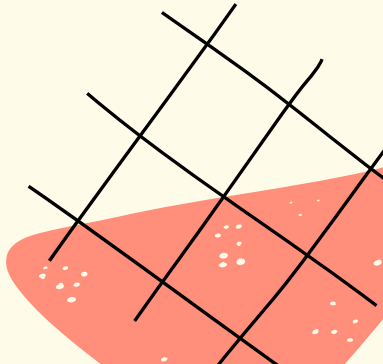
# Music Generation

## from selective emotion using Deep Learning

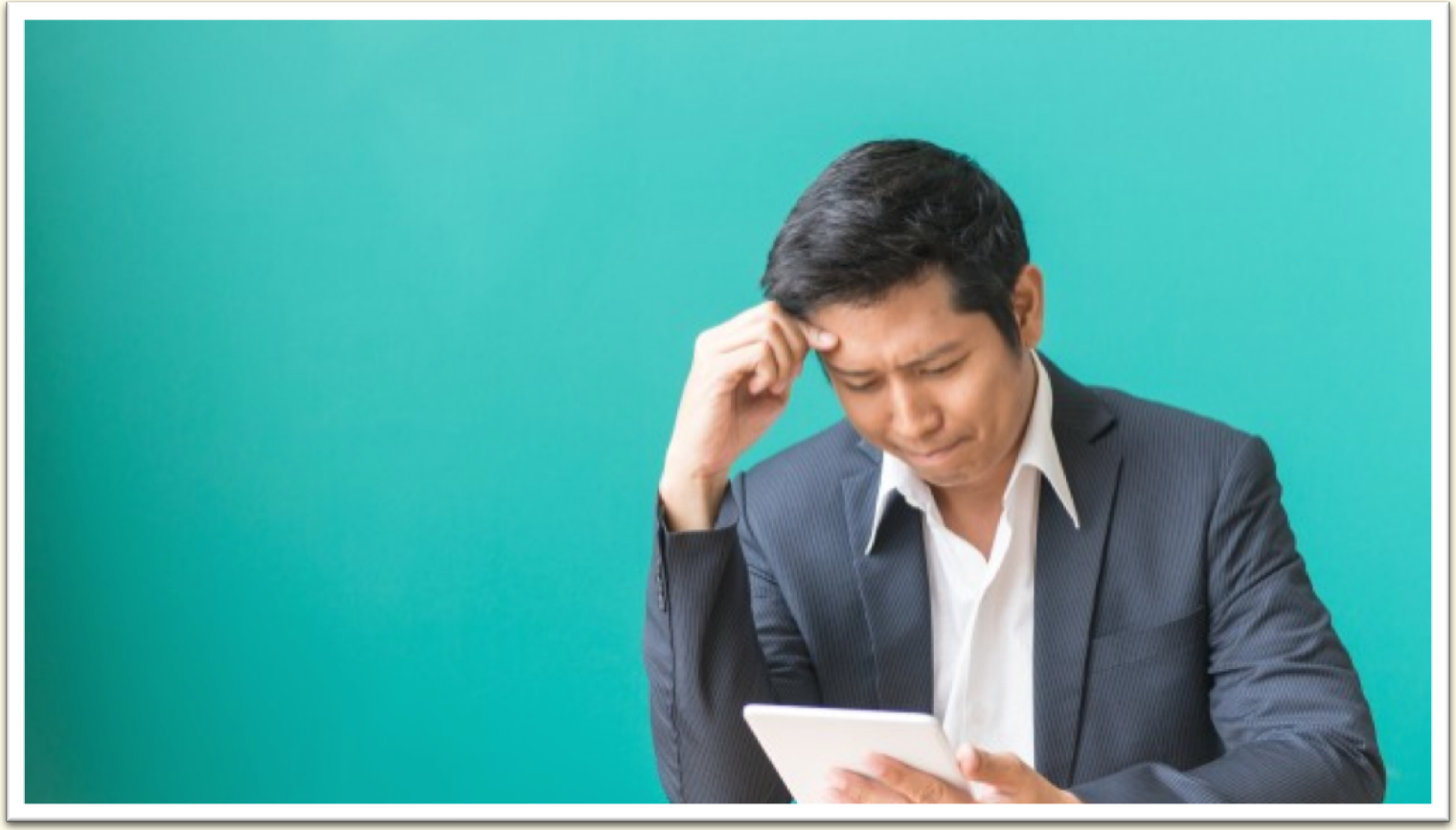


# CONTENTS

01

# The Problem - Solution

# PROBLEM



# After gathering the way to solve the problem

# Why we choose to generate music only Relax and Active songs?

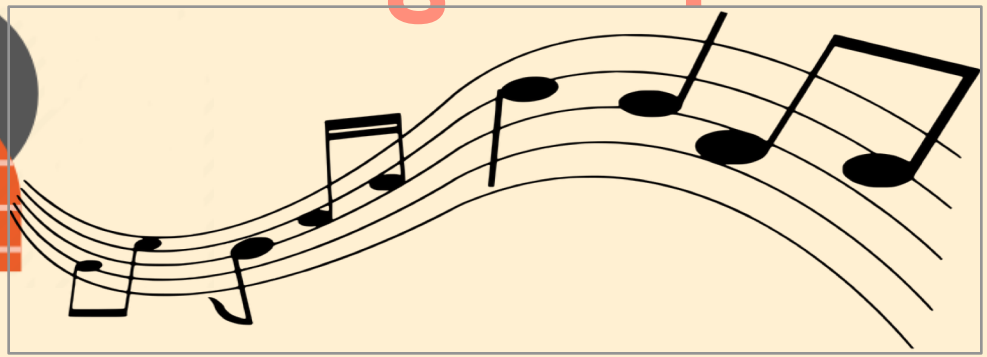
In this project, we focus on the emotional stress that why we choose to generate only relax and active. For other types of music such as sadness or happiness, our model can easily be generated those types of songs by adding new dataset of other types and some code.

02

# Our Data Product

# Music Generation

## from selective emotion using Deep Learning



# How it works

1

Choose type of generating music

Relax or Active

2

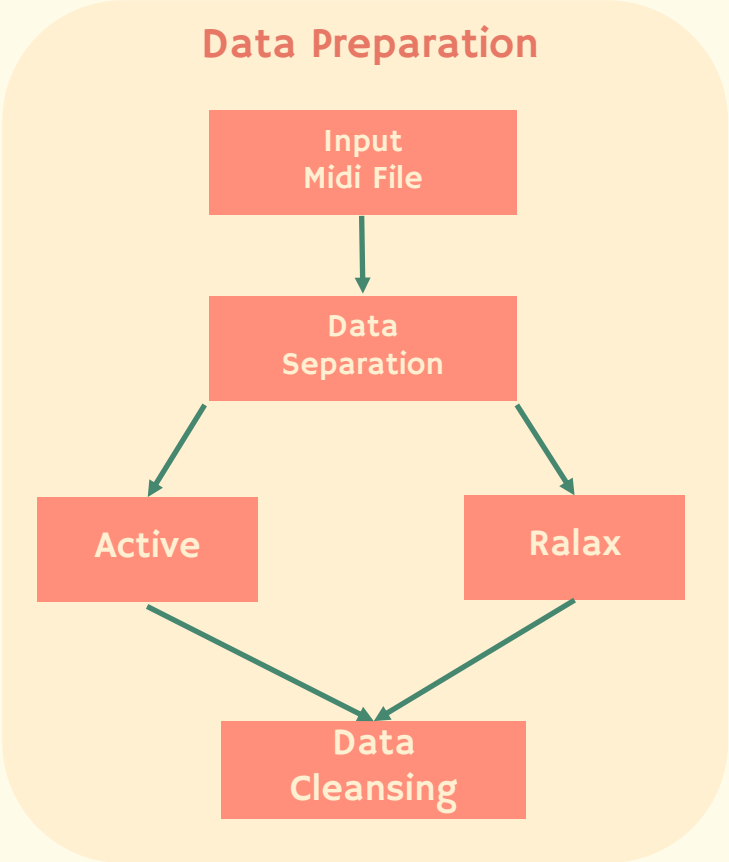
Generated 5 songs based on types of music that your chosen (1) via

Deep learning algorithms

03

# Our Algorithms

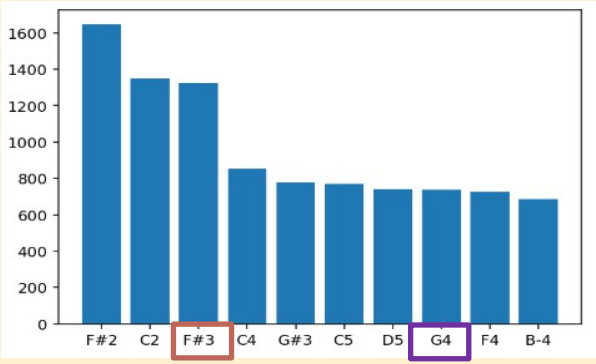
# Overview Algorithm



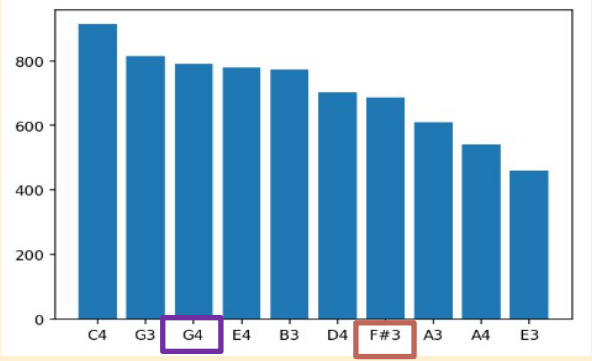
# Overview Algorithm

## Data Exploration

**Relax**




**Active**

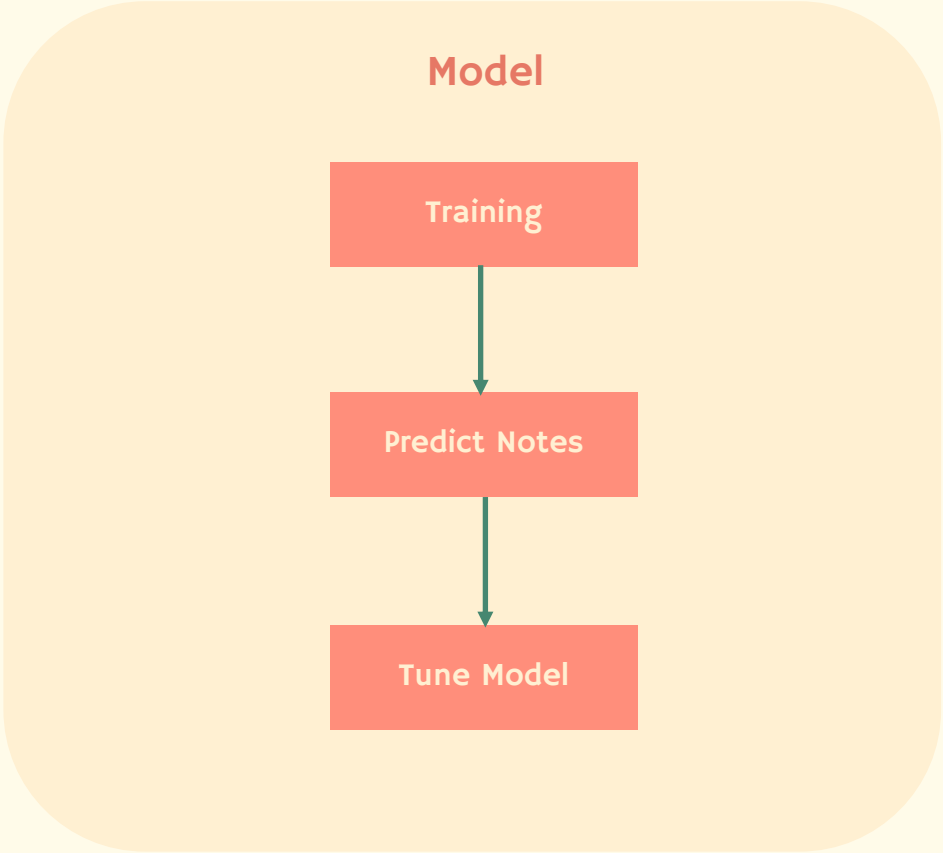



1. Rhythms , BPM (Beat per minute >> notes /60 min)
   - The higher, the more activeness.
   - The lower, the more relax
2. The Similarity between ground truth active or relax and our shingle notes
   - Find the most probability to be one of those types
3. The %Active Chord and %Relax Chord can be classify the type of music
4. Compare similarity of notes

   #similarity note 2 instrument in a point of time / #note
   #similarity note 3 instrument in a point of time / #note
   #similarity note 4 instrument in a point of time / #note
   #similarity note >4 instrument in a point of time / #note
   the more value , the more energic

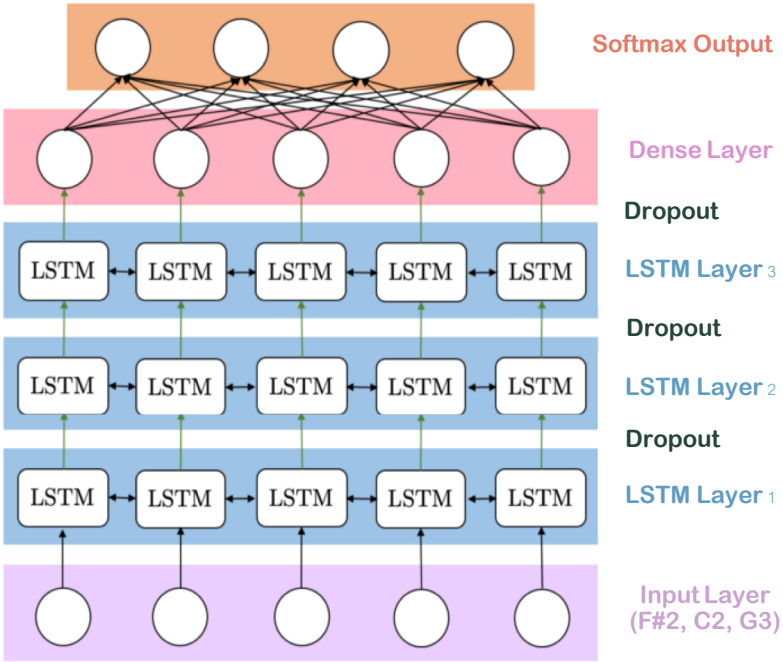
Then, using 4 of these mentioned plot as scatter plot and clustering.

# Overview Algorithm



# Model: LSTM

Model Architecture




## The details of LSTM model

**Dataset**: Midi dataset 74 songs

- Active: 42 songs
- Relax: 32 songs

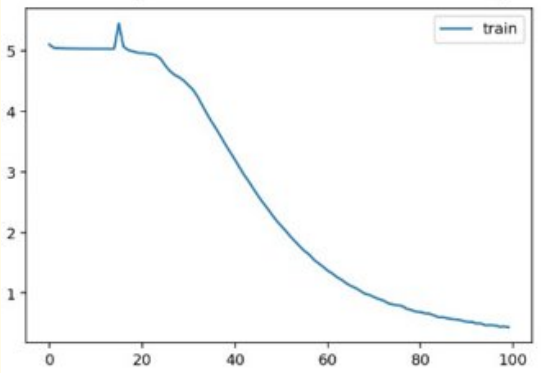
**Epochs:** 100
**Batch size:** 64
**Loss:** Categorical Cross Entropy

## Implementation

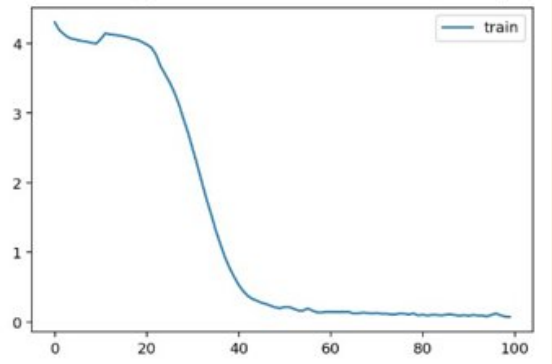
1. Separated dataset into 2 emotions that is active and relax
2. The model will get notes from songs in dataset that separated by emotion
3. Convert each note to integer and prepare note sequence before put into the model
4. Training LSTM model
5. Predict and convert integer back to note
6. Generated music

# Result

**Relax**

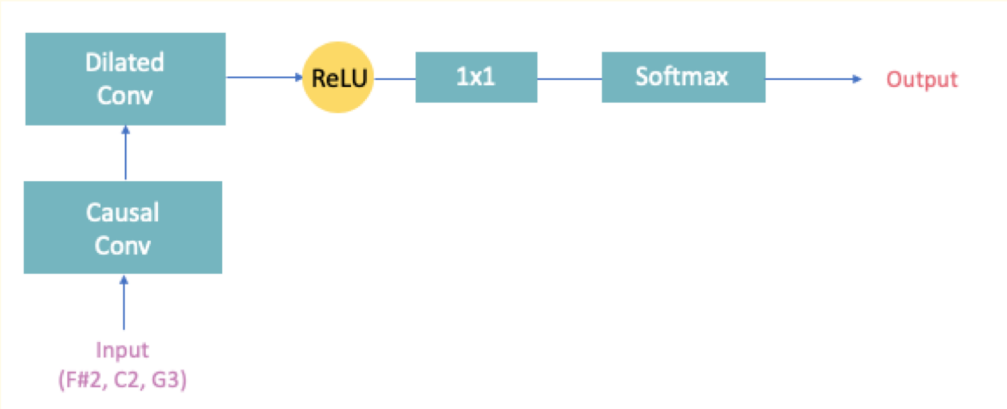



**Active**

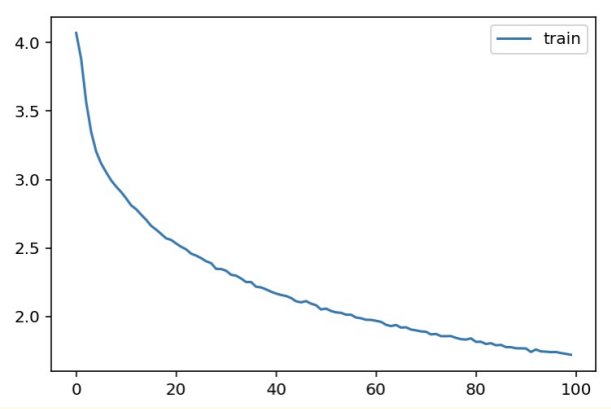


# Model: Simple WaveNet

## Model Architecture




## Result




**The details of WaveNet model**

**Dataset:** Midi dataset 74 song

- Active: 42 song
- Relax: 32 song

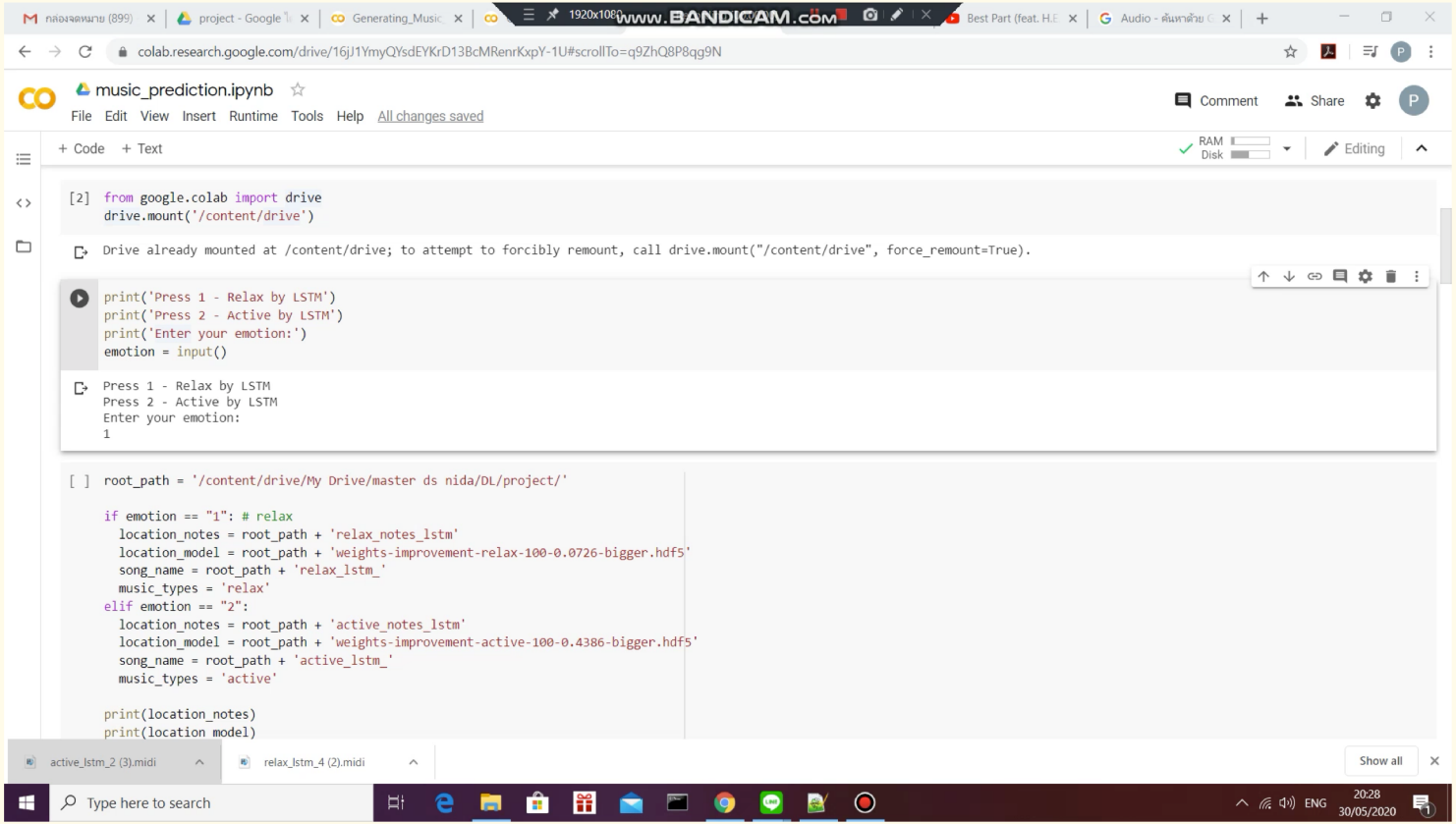
**Epochs:** 100

**Batch size:** 128

**Dilation rate:** 2, 4, 6

**Loss:** Sparse Categorical Cross Entropy

# Live Demo


colab.research.google.com/drive/16jJ1YmyQYsdEYKrD13BcMRenrKxpY-1U#scrollTo=q9ZhQ8P8qg9N
music_prediction.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[2] from google.colab import drive
drive.mount('/content/drive')
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
print('Press 1 - Relax by LSTM')
print('Press 2 - Active by LSTM')
print('Enter your emotion:')
emotion = input()
Press 1 - Relax by LSTM
Press 2 - Active by LSTM
Enter your emotion:
1
[ ] root_path = '/content/drive/My Drive/master ds nida/DL/project/'
if emotion == "1": # relax
location_notes = root_path + 'relax_notes_lstm'
location_model = root_path + 'weights-improvement-relax-100-0.0726-bigger.hdf5'
song_name = root_path + 'relax_lstm_'
music_types = 'relax'
elif emotion == "2":
location_notes = root_path + 'active_notes_lstm'
location_model = root_path + 'weights-improvement-active-100-0.4386-bigger.hdf5'
song_name = root_path + 'active_lstm_'
music_types = 'active'
print(location_notes)
print(location_model)
active_lstm_2 (3).midi
relax_lstm_4 (2).midi
Show all
Type here to search
ENG
20:28
30/05/2020