Semester Project

Generative AI for Music Based on Mood

Introduction

A music generation system employing LSTM neural networks which interprets user input to predict musical moods like Aggressive, Dramatic, Happy, Romantic, or Sad. Leveraging MIDI data from audio files, it generates musical notes, creating unique compositions.

Roles

Ali Awais Safdar	**Zainab Kashif**
Training and evaluation	**Training and Frontend design and UI management**
Huzaifa Liaqat	**Hunaina Ehsan**
Mood Classification	**Dataset acquisition and preprocessing**

Dataset Used

MUSIC_CLASSIFICATION DATASET

From Kaggle (8.96 GB)

Contains 5 folders of Aggressive, Dramatic, Sad, Happy and Romantic

Each folder contains .wav files of 4-5 seconds each

Each .wav file has a combination or the use of the specific notes for the type of music it is mainly used in (sad etc.)

Chosen Model

RNN model with LSTM layers is selected for mood to music generation for our system.
We used sub-class modeling to cater dynamic inputs.

Reason

Sequential Learning

Music often follows patterns over time, which LSTM networks are well-suited to understand

Handling Long-Term Dependencies

Defines a sequence length to capture dependencies in the music data over this window

Learning Musical Structure

The network's architecture, especially its memory cell, allows it to remember and forget information selectively

Sequence Prediction

LSTMs are effective in predicting the next element in a sequence based on the patterns it has learned

Model Training and Hyperparameters

The model architecture is based on a recurrent neural network (RNN) and consists of two LSTM layers.

Dropout layers with rates of 0.2 and 0.3 to prevent overfitting, also One Dense Layer with 256 Units which transforms the features learned.

The output layer employs the **softmax activation function**, offering a nuanced representation of probabilities for diverse generated output sequences corresponding to a given input.

```
...
260/260 [==============================] - 191s 737ms/step - loss: 4.5127
Epoch 3/200
260/260 [==============================] - 172s 662ms/step - loss: 4.4776
Epoch 4/200
120/260 [===========>.................] - ETA: 1:34 - loss: 4.4390
```

Model Training and Hyperparameters

Data Preparation

Input sequences and corresponding output sequences are prepared using the **prepare_sequences** function.
Which are used to employ a sliding window utilizing tuples of network_input and network_output necessary for the training of our model.

Model Training and Hyperparameters

Model Compilation and Summary

The optimization objective is governed by **Categorical Crossentropy loss**, ensuring effective model training for categorical outputs. The choice of the **Adam optimizer** facilitates efficient weight updates, enhancing the convergence and overall performance of the training process.

The model is constructed using the defined layers in the call method which implements its forward pass. Model is compiled and summarized using the **build_model** method.

Model Training and Hyperparameters

Training Process

Training is performed using the train method of the DynamicInputModel class.

Number of epochs = 200

Batch size = 128

Learning rate = 0.001 (Initially)

Model weights are saved during training using the **ModelCheckpoint callback**.

Best weights are saved based on minimizing the training loss.

As Adam optimizer changes learning rate dynamically after each epoch to make the weights converge faster.

Accuracy & Results

Objective Evaluation

For objective evaluation quantifiable metrics were computed to assess model performance on a particular category of music generation.

In this case we assessed music generated by 'Happy' mood and compared it to our 'Notes_Happy' file, obtained from the 'dataset_midi/happy' subdirectory which is used to calculate the input, output pairs for the training of our model.

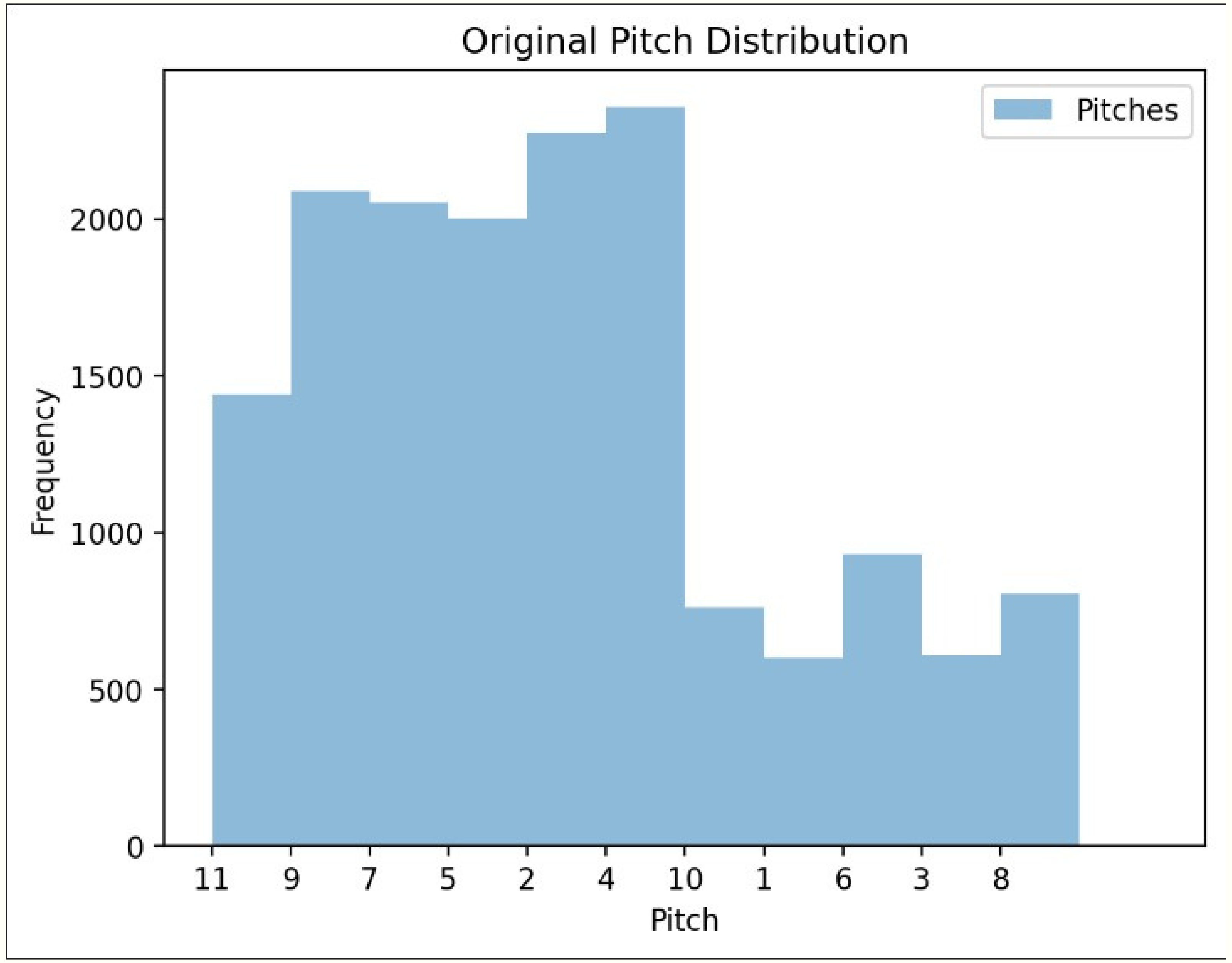
Jaccard Similarity

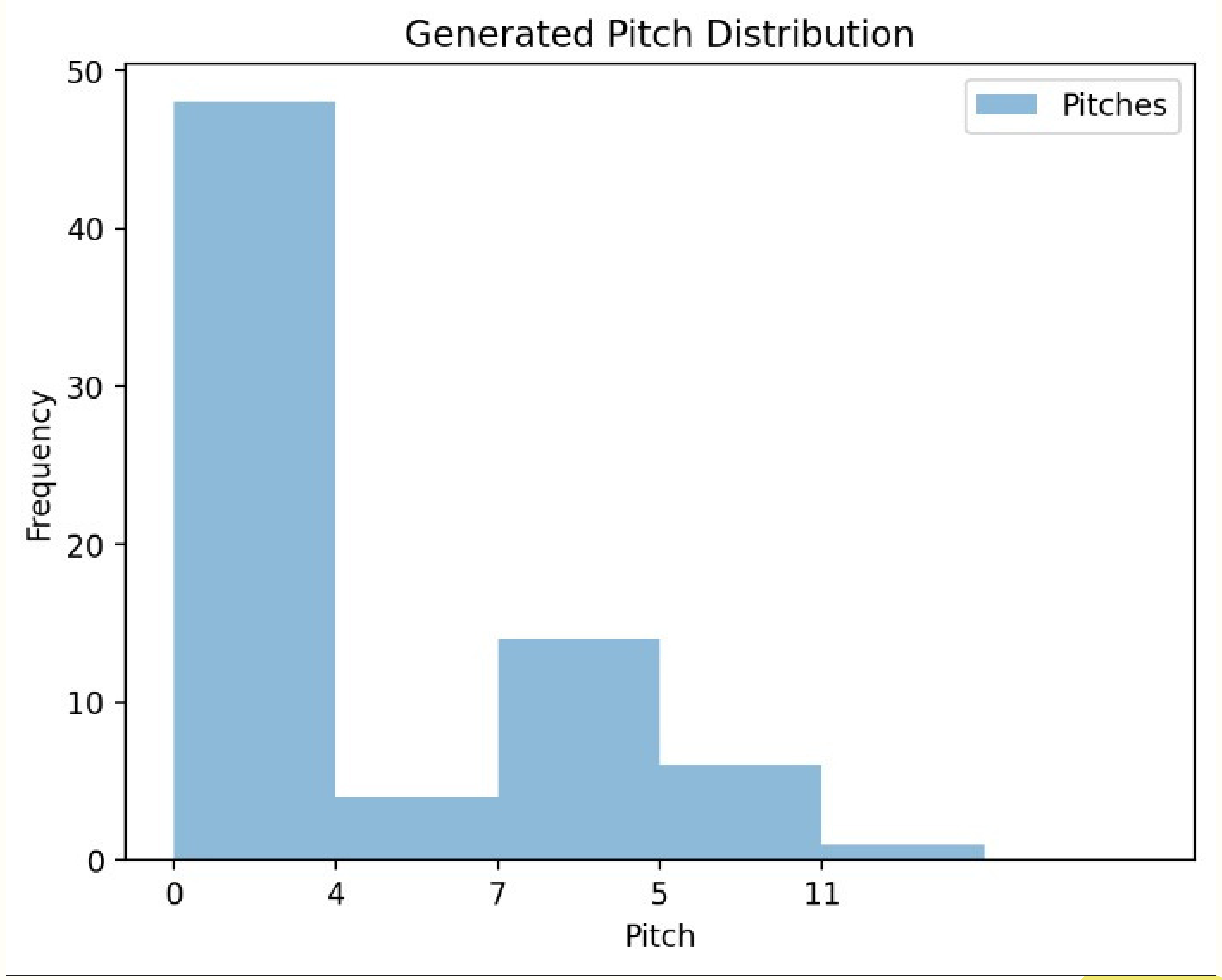
```
Jaccard Similarity for Pitches: 0.5
```

The **Jaccard similarity** ranges from 0 to 1, where 0 indicates no overlap, and 1 indicates complete overlap. A similarity of 0.5 suggests that half of the pitches in the two sets are common, while the other half is unique to each set.

Original versus Generated Pitch Distribution

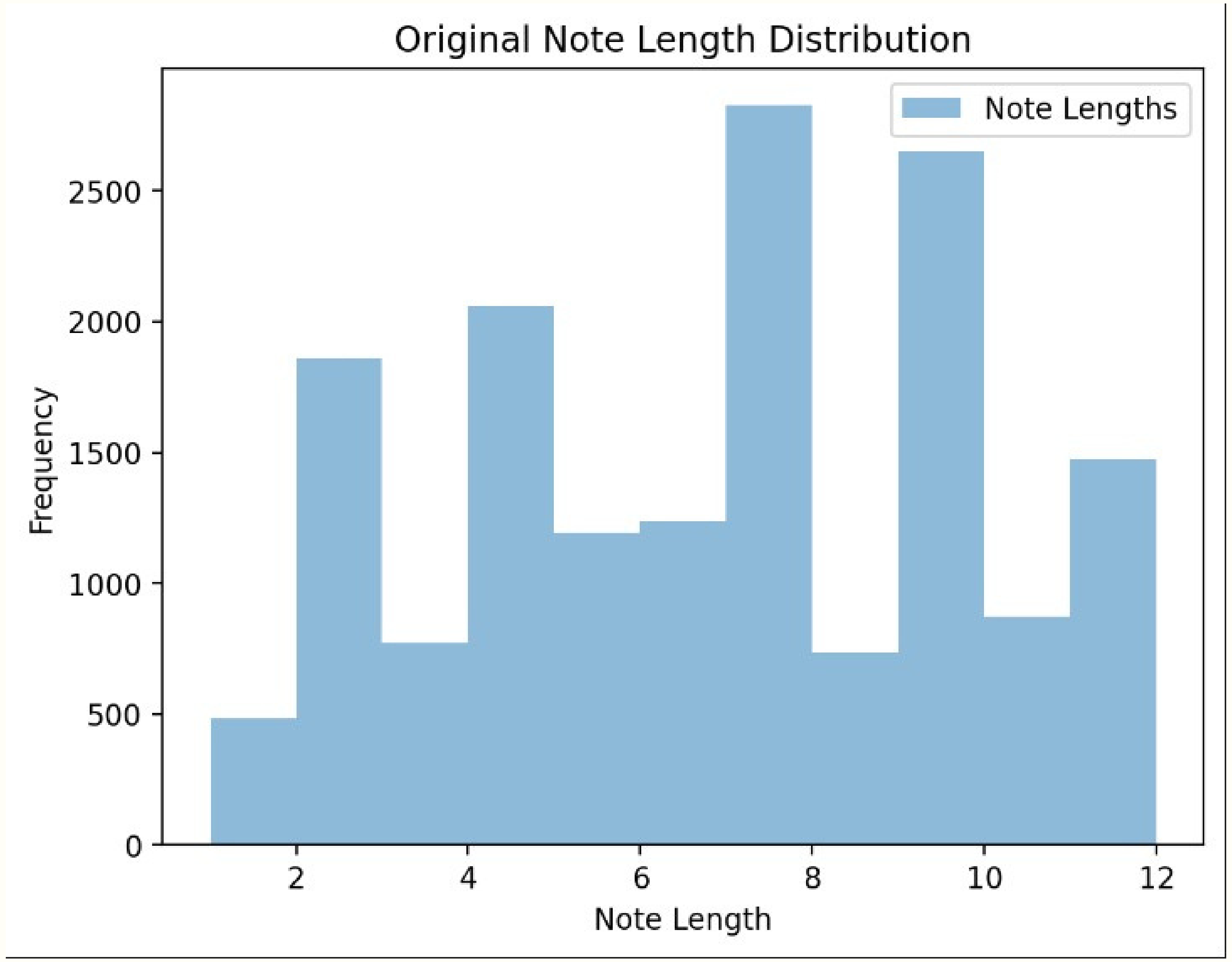
In the original composition, a rich assortment of pitches is thoughtfully distributed, contributing to a diverse and engaging musical experience. However, in the generated music, where the model draws on random input sequences tied to a specific mood, there's a notable tendency for one particular pitch to dominate, resulting in a more focused but potentially less varied musical output.

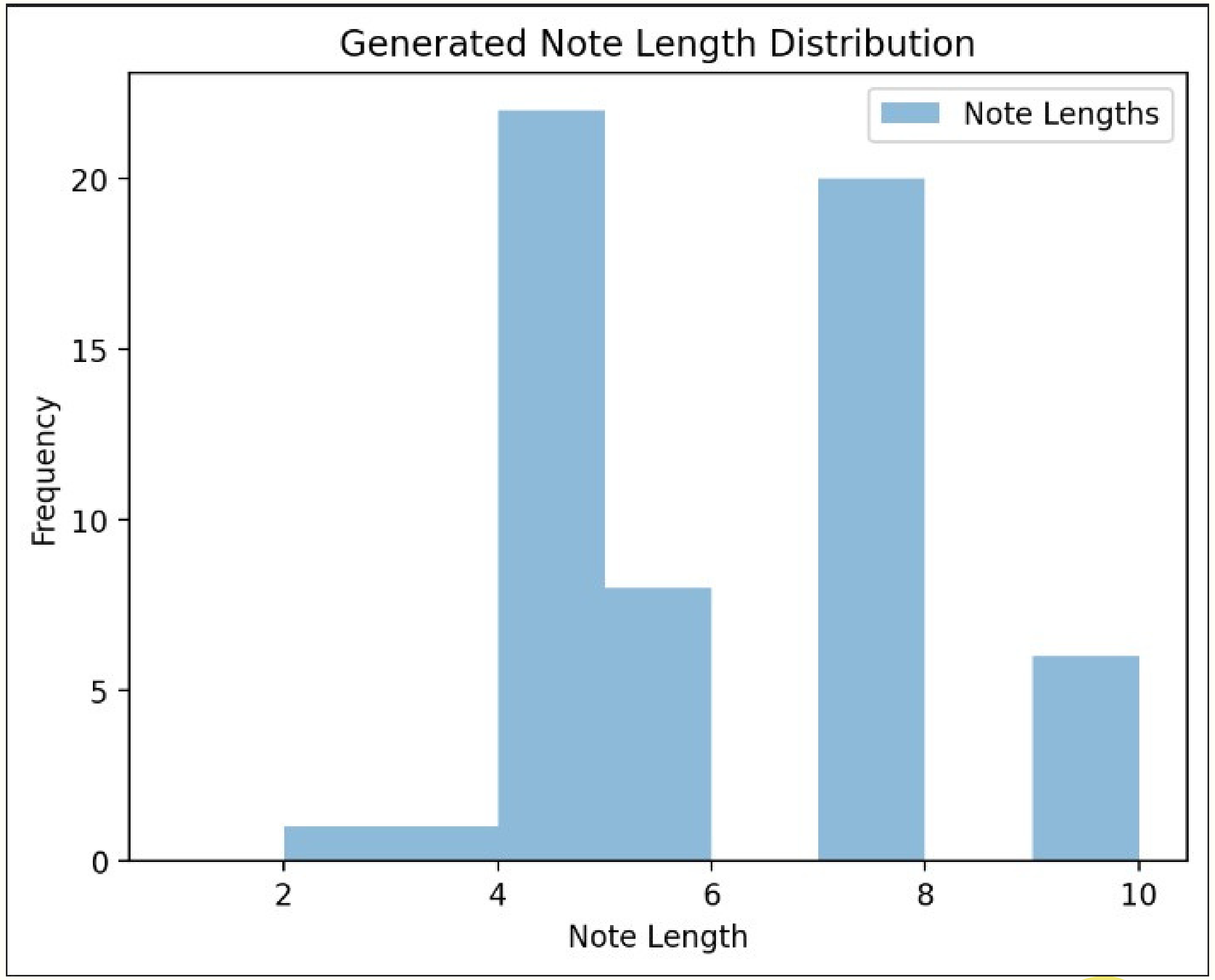




Original versus Generated Notes Distribution

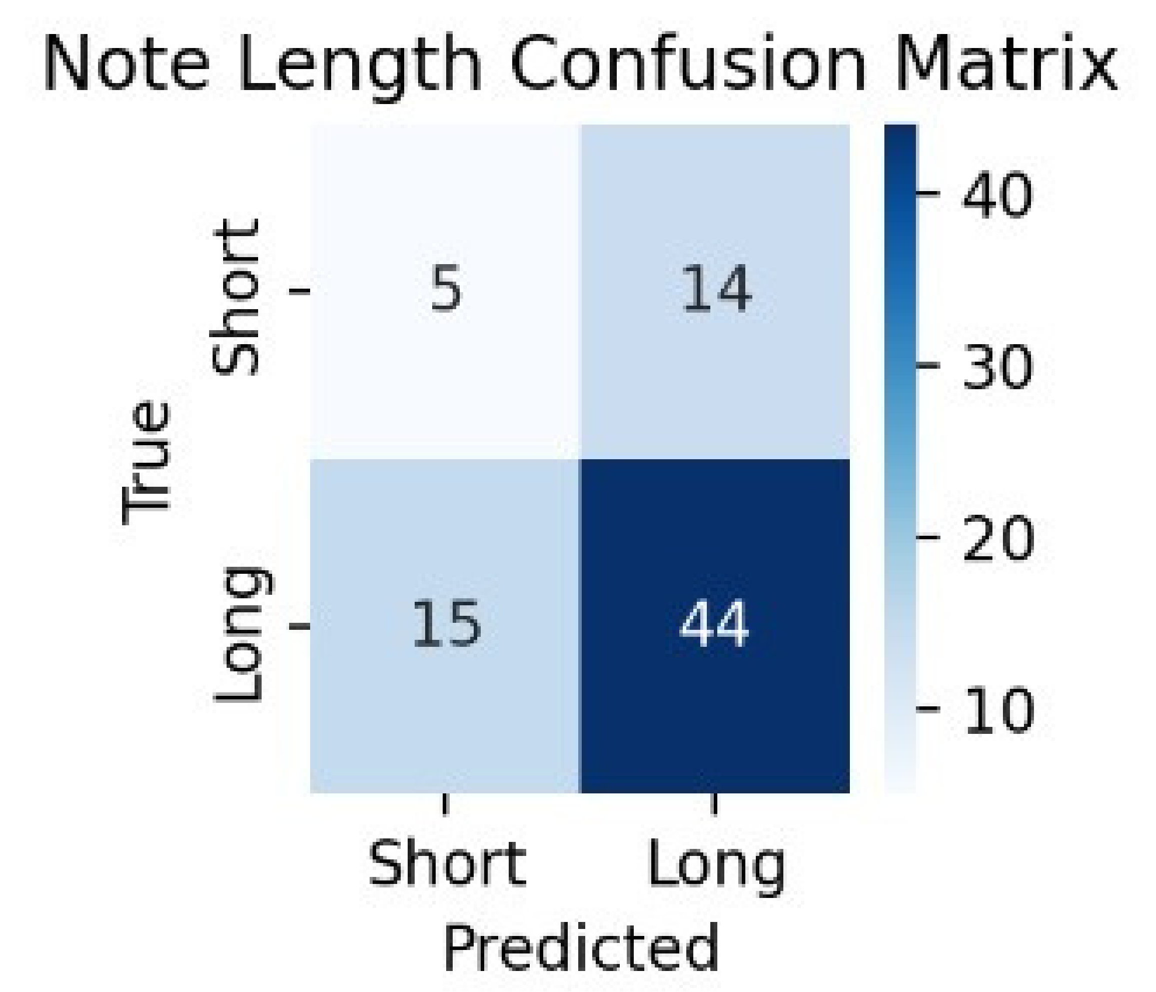
In the original composition, we observe a well-distributed bins of notes with varying lengths, creating a harmonious balance. However, in the generated piece, a limited set of notes is predominantly used in succession, resulting in a composition characterized by fewer alternating notes and a tendency towards repetition in the musical arrangement.





Confusion Matrix

The confusion matrix indicates the model's proficiency in classifying note lengths as "short" or "long." Impressively, it accurately identifies 40 short and 44 long notes. However, there's a slight challenge with short notes, as 14 are mistakenly classified as long. Overall, the model excels but shows room for improvement in distinguishing shorter musical elements.



F1 Score

```
F1 Score: 0.6313828621520928
```

An **F1 score** of 0.63 suggests a moderate level of performance in balancing precision and recall. Our model is achieving a reasonable trade-off between minimizing false positives and false negatives.

Subjective Evaluation

For the Subjective Evaluation phase, we implemented a Google Form where team members engaged diverse users to independently utilize our model for generating music across various moods. Participants then listened to the generated music and rated their satisfaction on a scale of 0-10, providing valuable subjective insights into the quality and user experience of the music produced by our model.

Subjective Evaluation

Subjective evaluation is done by gathering feedback as rating from the users of our system. The **average rating is 7.57** which indicates generally positive feedback from users. On average the Users, find the generated music to be satisfactory.

Conclusions

Difficulties Faced

1. Dataset Limitations (Duplications)
2. Subjective Evaluation
3. Data Preprocessing
4. Model Complexity

Limitations

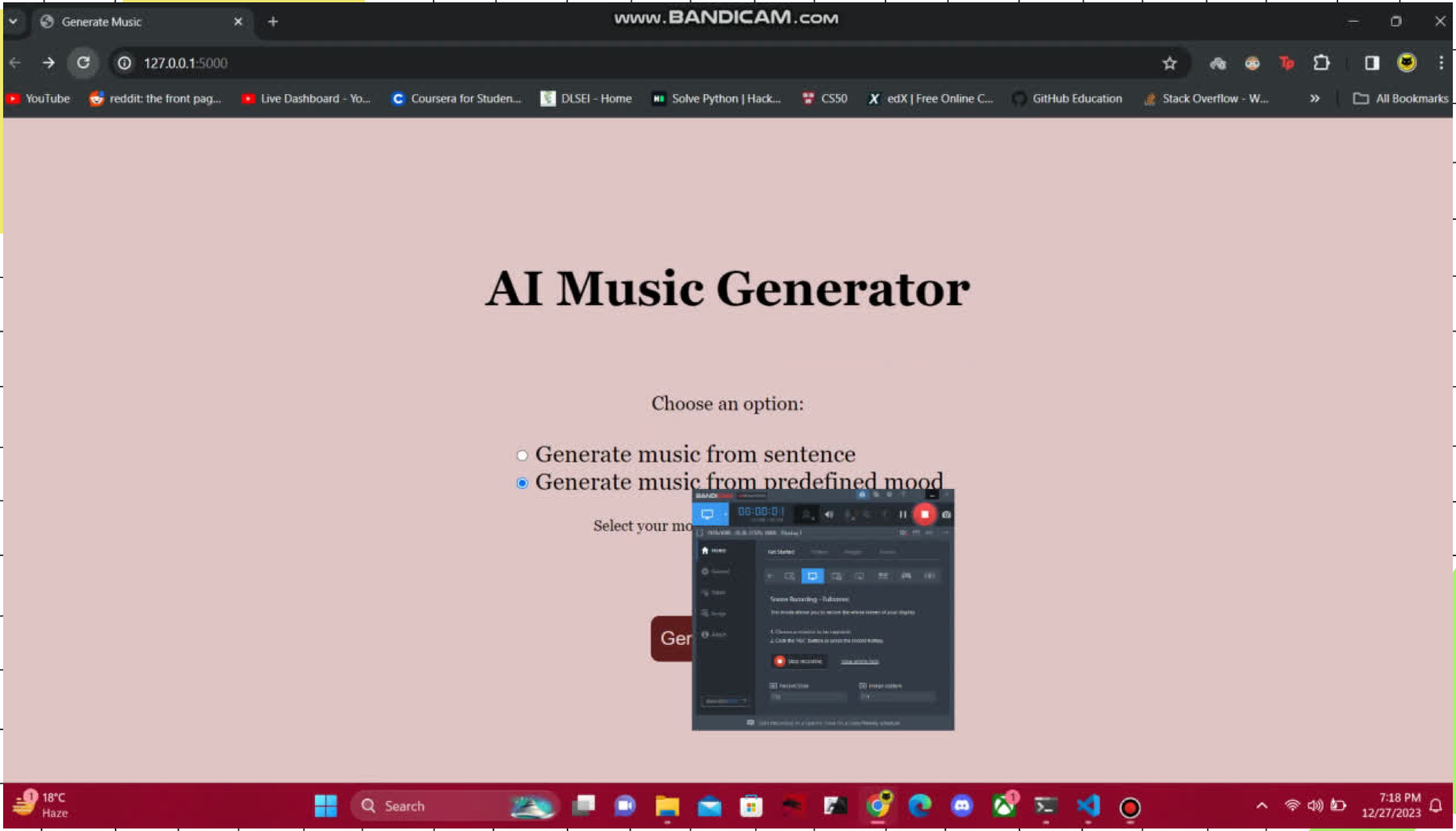
1. Limited Music Composition Styles
2. Dependency on MIDI Format

Future Improvements

1. Dataset Expansion
2. Incorporate User Preferences:
3. Explore Transfer Learning
4. Experiment with Different Architectures

Demo

[Demo Link]()


Generate Music
www.BANDICAM.com
127.0.0.1:5000
YouTube
reddit: the front pag...
Live Dashboard - Yo...
Coursera for Studen...
DLSEI - Home
Solve Python | Hack...
CS50
edX | Free Online C...
GitHub Education
Stack Overflow - W...
All Bookmarks
AI Music Generator
Choose an option:
Generate music from sentence
Generate music from predefined mood
Select your mo
Ger
18°C
Haze
Search
7:18 PM
12/27/2023

Thank you