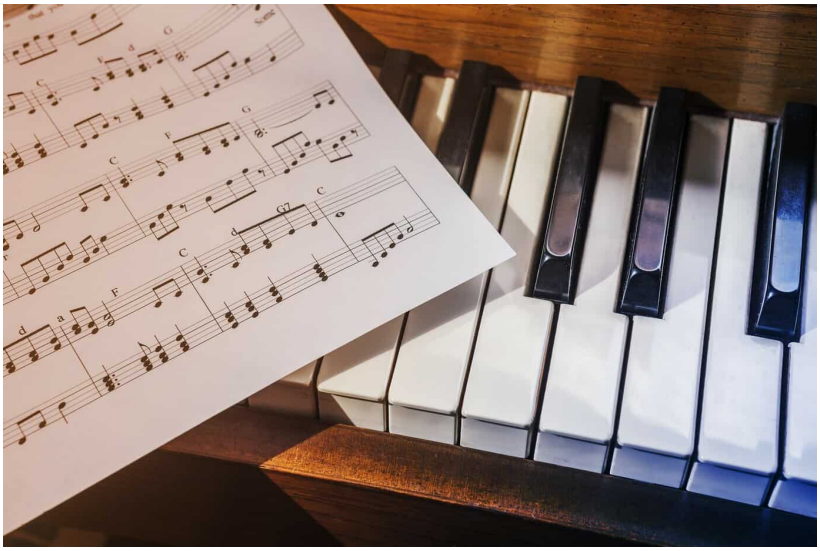
REPORT ON

KAGGLE :

MUSIC GENERATION - LSTM

https://www.kaggle.com/code/karnikakapoor/music-generation-lstm/notebook



BY :

Aman Srivastava - MT21007
Mahak Sharma - MT21047
Rishabh Kumar Pundhir - MT21071
Rutvikkumar Bandhaniya - MT21116

For this project, I will be using MIDI files of classical piano music. The dataset includes various artists. I will be working with famous artists - Frédéric Chopin's and Beethoven's compositions one by one.

IMPORTANT : Switch to GPU runtime if using google colab

The below work is on Chopin's data.

Total files - 48
Duration of files - 30 seconds to 9 minutes

LOADING THE DATA

- First of all, I made a list of all the songs in the Chopin's folder **parsed as music21 stream.**
- Then I will be creating a function to **extract chords and notes** out of the data creating a corpus.

  Note: The musical notes are the building blocks of the music. It pertains to a pitch associated with a specific audio vibration. Western music utilizes twelve musical notes.

  Chord: A group of notes that sound good together is a chord.

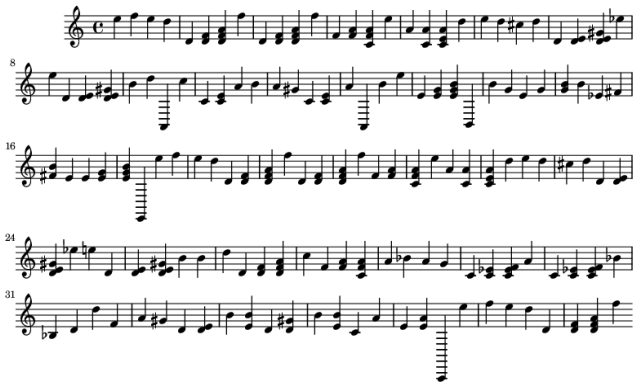
  `Total notes in all the Chopin midis in the dataset: 63429`

  So we have our data in the form of a corpus. A list of strings.Each string indicates a musical note.

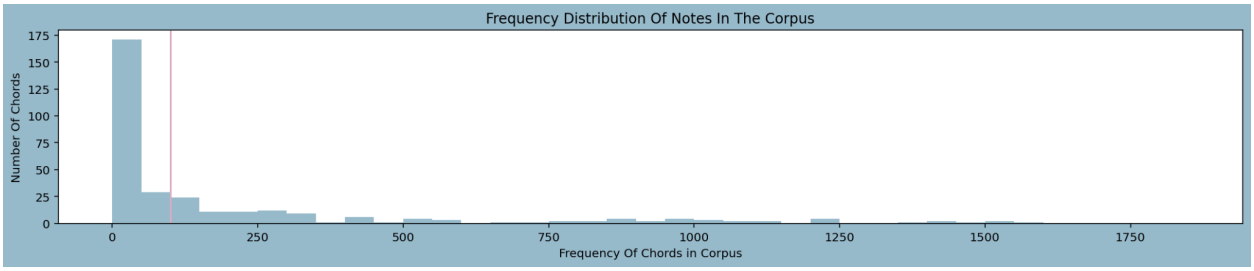
DATA EXPLORATION

- **`First fifty values in the Corpus: ['E5', 'F5', 'E5', 'D5', '2.5.9', 'F5', '2.5.9', 'F5', '5.9.0', 'E5', '9.0.4', 'D5', 'E5', 'D5', 'C#5', 'D5', '2.4.8', 'E-5', 'E5', '2.4.8', 'B4', 'D5', 'A2', 'C5', '0.4', 'A4', 'B4', 'A4', 'G#4', '0.4', 'A4', 'A2', 'B4', 'E5', '4.7.11', 'B2', 'B4', 'G4', 'E4', '7.11', 'B4', 'E-4', '6.11', 'E4', '4.7.11', 'E2', 'E5', 'F5', 'E5', 'D5']`**

- Sample music piano sheet from Chopin's data corpus :



- **Total unique notes in the Corpus: 317**
- **Average recurrence for a note in Corpus: 200.09148264984228**
  Most frequent note in Corpus appeared: 1869 times
  Least frequent note in Corpus appeared: 1 time



- **Total number of notes that occur less than 100 times: 200**
- **Removed notes that were played less than 100 times**. I mean, if Chopin liked them he would have played it a lot more often.
  Length of Corpus after elimination the rare notes: 59853

DATA PREPROCESSING

- **Creating a dictionary:** Creating a dictionary to map the notes and their indices. We have the note's name as a string, the Corpus. For the computer, these names are just a symbol. So we create a dictionary to map each unique note in our Corpus to a number. And vice versa to retrieve the values at the time of prediction. This will be used to encode and decode the information going in and getting out of the RNN.

  Dictionary/mapping : {'A1': 1, 'A2': 2, 'A3': 3, 'A4': 4,'A5': 5,'A6': 6,'B-1': 7,'B-2': 8,'B-3': 9,'B-4': 10, ...} *in sorted order*

  Total number of characters: 59853

  Number of unique characters: 228

- **Encoding and Splitting the corpus:** Encoding and splitting the corpus into smaller sequences of equal length(40 in our case): At this point, the Corpus contains notes. We will encode this corpus and create small sequences of equal lengths(40) of features and the corresponding targets. Each feature and target will contain the mapped index in the dictionary of the unique characters they signify.

Feature Target

E.g. Let Coprus = [A1, C3, E1, D2, B1, B2, A3, D1, E5, C2 .....] and Splitting Length = 3

Features = [[A1, C3, E1], [C3, E1, D2], [E1, D2, B1], [D2, B1, B2], [B1, B2, A3], [B2, A3, D1], ...]

Targets = [D2, B1, B2, A3, D1, E5, ...]

Total number of sequences in the Corpus: 59853-40 = 59813

- **Assigning X and y:** The labels are then resized and normalized. Whereas the targets are one-hot encoded.

  X.shape = (59813, 40, 1) Y.shape = (59813, 228)

- **Splitting Train and Seed datasets:** To create music we will need to send some input to the RNN. For that, we will set aside a part of the data as seeds. *We could have trained it all but I am no musician to come up with an input seed value.*

  Split ratio = 80:20

MODEL BUILDING

- Data is ready to be sent to the RNN for the training, but before that let us build the RNN model.

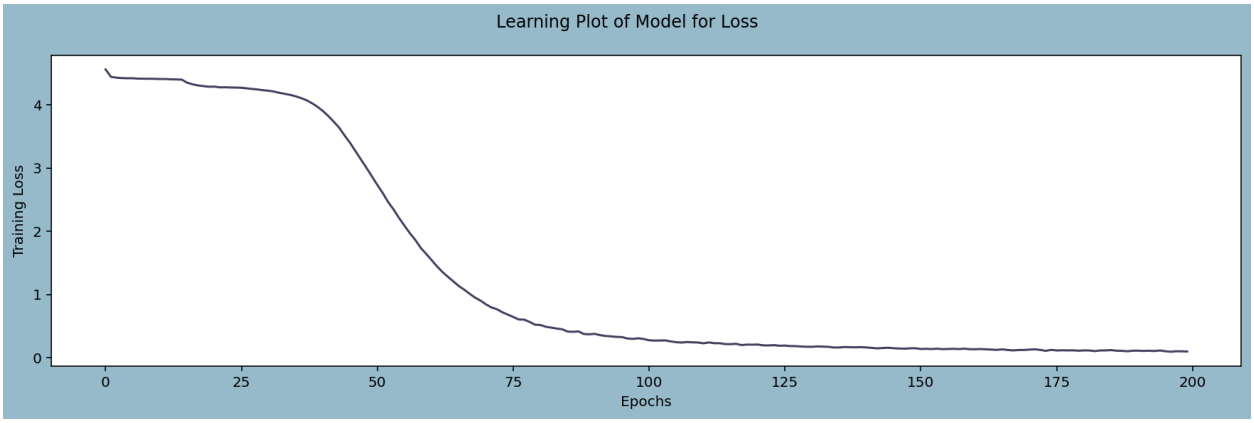
```
#Initialising the Model
model = Sequential()

#Adding layers
model.add(LSTM(512, input_shape=(X.shape[1], X.shape[2]), return_sequences=True))
model.add(Dropout(0.1))
model.add(LSTM(256))
model.add(Dense(256))
model.add(Dropout(0.1))
model.add(Dense(y.shape[1], activation='softmax'))

#Compiling the model for training
opt = Adamax(learning_rate=0.01)
model.compile(loss='categorical_crossentropy', optimizer=opt)
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 40, 512)           1052672

 dropout_4 (Dropout)         (None, 40, 512)           0

 lstm_5 (LSTM)               (None, 256)               787456

 dense_4 (Dense)             (None, 256)               65792

 dropout_5 (Dropout)         (None, 256)               0

 dense_5 (Dense)             (None, 228)               58596
=================================================================
Total params: 1,964,516
Trainable params: 1,964,516
Non-trainable params: 0
_________________________________________________________________
```

- **For fitting the model, batch_size=256, epochs=200**
- **Training time :**
  - **Around 7 hours without GPU runtime at colab.**
  - **Around 30 min using GPU runtime**

MODEL EVALUATION

- To evaluate my model, I shall be having a look at:
  1. The performance of the model via Learning Curves
  2. The melody created
- Plot of training loss



GENERATING MUSIC

- **Generating Music:** For generating the music, we are randomly selecting one sequence from X_seed and predicting its next note. Then this newly predicted note is appended to the sequence taken from X_seed and removed the first element/note from that sequence. Now this new sequence is used for predicting the next note and the next note will be appended to the sequence and the first note will be removed. This process will continue until the number of notes that user has specified.
- **Generated Sample 1** **Generated Sample 2**