

“CAPSTONE PROJECT: [A.I. MUSIC GENERATION”](http://en.wikipedia.org/wiki/Colpitts_oscillator)

**Mini-Project Report on**

**-:LYRICS GENERATION USING:-**

**-:LSTM MODELS AND MIDI: -FILES**



# By

# Ankit Kumar Srivastava C.S.E. – 1809710016

**Under the Supervision of**

**Mr. T. Akilan**

**Mr. Mayank Dixit**

A Project submitted to the Computer Science Department

In partial fulfillment of the requirements for the degree of B.Tech (COMPUTER SCIENCE ENGINEERING)

Galgotias College of Engineering and Technology ,Greater Noida ,U.P.

**India-201306**

**Affiliated to**



**Dr. A.P.J. Abdul Kalam Technical University**

**Lucknow , Uttar Pradesh,**

**India-226031**

May, 2020

# ABSTRACT

This is the report of my fifth semester mini-project of the mini-project lab. This report will give the reader an overview about the steps which are involved in the making of my capstone project lyrics generator with the help of Python, Deep Learning and various data science modules of Python. The project is about generation of lyrics with the help of various MIDI (Musical Instrumental Digital Interface) files. In modeling the project, various python modules are used for Data acquisition, Data cleaning, Exploratory analysis, Creating the Deep learning model, etc. Text generation, code generation, novel generation everything can be done using Markov chain model of ML or LSTM(DL model based on RNN).What I have done is I have taken various midi files of piano and combined its notes and chords and obtained them in the form of text after that I used those text mappings to feed the stacked LSTM RNN model (Prepared Sequential Data for LSTM). Network input has to be normalized before giving it as an input to LSTM RNN network. Network output are the classes which are encoded into one hot vector. Training took a bit of more time so I trained the model on google colab and saved the new\_weights.hdf5 model file. After having output notes list as prediction it can easily be converted to a midi file with certain timestamps to have some lyrics.

For making the project python modules like music21, glob, pickle, keras, etc had been used. For working with midi files music21 module was useful for slicing and editing the audio files.

The very special feature of my project is in future this kind of audio mixing can be done using midi files of different instruments like piano + guitar + brass + band in order to create pop music which would be really exciting. But training has to be done over a large dataset and over 1000 epochs to get a nice validation accuracy.

**Table of Contents**

Chapter 1…………………………………...

[INTRODUCTION](#_bookmark0)

[Musical Lyrics Generator](#_bookmark1)

[Overview](#_bookmark2)

* 1. [Purpose of the project](#_bookmark3)
  2. [Applications of the Project](#_bookmark4)
  3. [Summary :](#_bookmark5)

[Chapter 2](#_bookmark6)

[LITERATURE REVIEW](#_bookmark7)

* 1. [Related Technologies](#_bookmark8) 
     1. [Special modules available](#_bookmark9)
  2. [Project Information](#_bookmark10)

[Developer side view](#_bookmark11)

* 1. [Related Studies](#_bookmark12)
  2. [Summary:](#_bookmark13)

[Chapter 3](#_bookmark14)

[TOOLS AND TECHNIQUES](#_bookmark15)

* 1. [Software used with technical specifications](#_bookmark16) 
     + [Jupyter(the editor for python) and its versatile ..….………………..](#_bookmark17)
     + [pip and its command line specialty](#_bookmark18) ..
     + Modules the tools that inspire……………………………………....
     + [music21………………..](#_bookmark19) ..
     1. [Keras](#_bookmark20) …………….
     2. [Numpy...](#_bookmark21)
     3. [Other Data Science modules](#_bookmark22)

Fun with Lyrics Generator : …………………….............................................

* 1. [Summary:](#_bookmark28)

[Chapter 4](#_bookmark29)

[METHODOLOGIES](#_bookmark30)

* 1. [Design of the Algorithms](#_bookmark31)
  2. [Analysis procedures](#_bookmark32)
  3. [Implementation procedure](#_bookmark33) 
     1. [Details about hardware](#_bookmark34)
     2. [Details about software/ algorithms](#_bookmark35)
  4. [Verification of functionalities](#_bookmark36)
  5. [Details about simulation :](#_bookmark37)
  6. [Summary:](#_bookmark38)

[Chapter 5](#_bookmark39)

[RESULTS AND ANALYSIS](#_bookmark40)

* 1. [Presentation of the findings](#_bookmark41) 
     1. [Software results](#_bookmark43)
  2. [Discussion of the findings](#_bookmark44) 
     1. [Comparison with initial GOAL](#_bookmark45)
  3. [Recommendations](#_bookmark46)

[Summary:](#_bookmark47)

[Chapter 6](#_bookmark48)

[CONCLUSION](#_bookmark49)

[REFERENCES](#_bookmark50)

Chapter 1

## INTRODUCTION

## 

## Random Lyrics Generator using LSTM(RNN) :

Lyrics generator is a Deep Learning Model which takes input audio files of instruments in the midi format not in any other like mp3 or mp4 and by using RNN model it predicts a new set of lyrics combination. Using my this project one can easily generate a combination of melodious song which consists of audio of various instruments. It is basically and application of sequential RNN(LSTM) model. It can easily be used in opera musics and places where audio mixing is of great importance. Sometimes it is really difficult to think of a background sound for a specific occassion. Using my this project one can easily perform audio mixing upto a certain level with proper change of melody every time you train the model. Hyperparameter tuning and training of the model upto 150-200 epochs can result in better validation accuracy.

## Overview:

Lyrics generator does audio mixing in simple terms giving different melodies by hyper parameter tuning. This project gives the exact feeling of using a real audio mixing software with the exception that it cannot process mp3 or mp4 files but midi files of the instruments. Hence simple instrumental mixup can be obtained using this generator which is quite awesome.

## Purpose of the project:

As a student the project has helped me a lot to deep dive into recurrent neural networks. Its basic purpose is audio mixing with different melodies of instrument.

## Applications of the Project:

For the musicians my this project could be really helpful tool. As it can give nice gist of the instrumental audio of different instruments when played together.

## Summary:

In this chapter I explained about the project and its applications and the theory of project and little introduction about how to use it efficiently. Also explained about the special feature of the project . As so far the project does not have any user interface which would be good for a newbie, I would advice any one going for the project to install Jupyter notebook or go to Google Colab. So that you do not face any issue with the module absence and be able to use it easily and train it.

# Chapter 2 LITERATURE REVIEW

## Related Technologies.

**Artificial intelligence:**

Artificial intelligence is the most growing field in the current era. It is the intelligence demonstrated the machines which they gain through experience. The neural networks specially help them to develop this intelligence. Python now a days provides a nice platform to communicate with the machines and make them to perform the required task. Its inbuilt as well as third party modules are very helpful in making useful projects at the industry level. I have demonstrated a small gist of the power of Python programming language through this project which made me to understand its third party modules nicely.

## Project Information:

**Inroduction:**

I have implemented the melody tone mixup using piano tone files which I obtained from Kaggle. There are octaves in Piano. These octaves are repeated on the left and the right side of the piano with increasing frequency on the right side and a low frequency on the other side. A note is the unit of sound in piano melody whereas there are 5 accidental notes. Hence there are total of 12 notes in an octave. A chord is equal to multiple notes at the same time. Hence I preprocessed the piano midi audio data to perform an audio mixup as it was easily available on Kaggle.

**Music21 module for processing midi files:**

Music21 python module is very much useful module for manipulating and preprocessing midi audio files in python. It can return a midi object which contains notes, chords and other components. music21 can be used for audio to text extraction in the form of notes. It results in notes+chords+offsets.

The are sections and subsections in music21 library which contains several notes and chords on loading a midi audio file. Now audio file in the form of text can be converted into flat list with an iterator on the list. Some chords may have two or more notes. From here time offset of a particular note can also be detected.

## Parsing the midi files:

Now the parsing of file is done and note and chords are detected. And finally sequential data is prepared for the RNN model with sequential length of 100. In my case there are 359 classes of distinct notes. ML or DL model doesn’t work with string data hence there is need of mapping elements of notes with integer value by creating a python dictionary. We try to create a sequential normalized data which is supervised, from the initial list of notes using a stride of one. Since the network output are classes we need to one-hot encode the classes into vectors. Now input becomes easily feedable into the RNN(LSTM) model.

## 

## Parsing the midi files:

Now the Sequential Model is created using Keras Modules and Tensorflow Backend with RNN layers, proper dropouts and Dense layer at the end. The model is compiled and trained over 100 epochs on Google colab and a checkpoint is created wherever a nice validation- accuracy of the model is detected. After that just load the hdf5 file of the trained model and go for well trained weights in the program.

## Predicting new Sequence:

Now from the output list of predictions by creating reverse dictionary which maps stringified notes to integers. Taking argmax of the probability of a single prediction we get the required class number and after that we map the required integer to the notes. Finally with a offset of 0.5 combine all the notes in the list to get a finite melodious tone. All the tones will be played for short and fixed (0.5) duration of time.

Now create a Stream object using music21 and store the generated lyrics on the hard disk. Hurray!! Now it can be shared and you can receive your feedback.

## Summary:

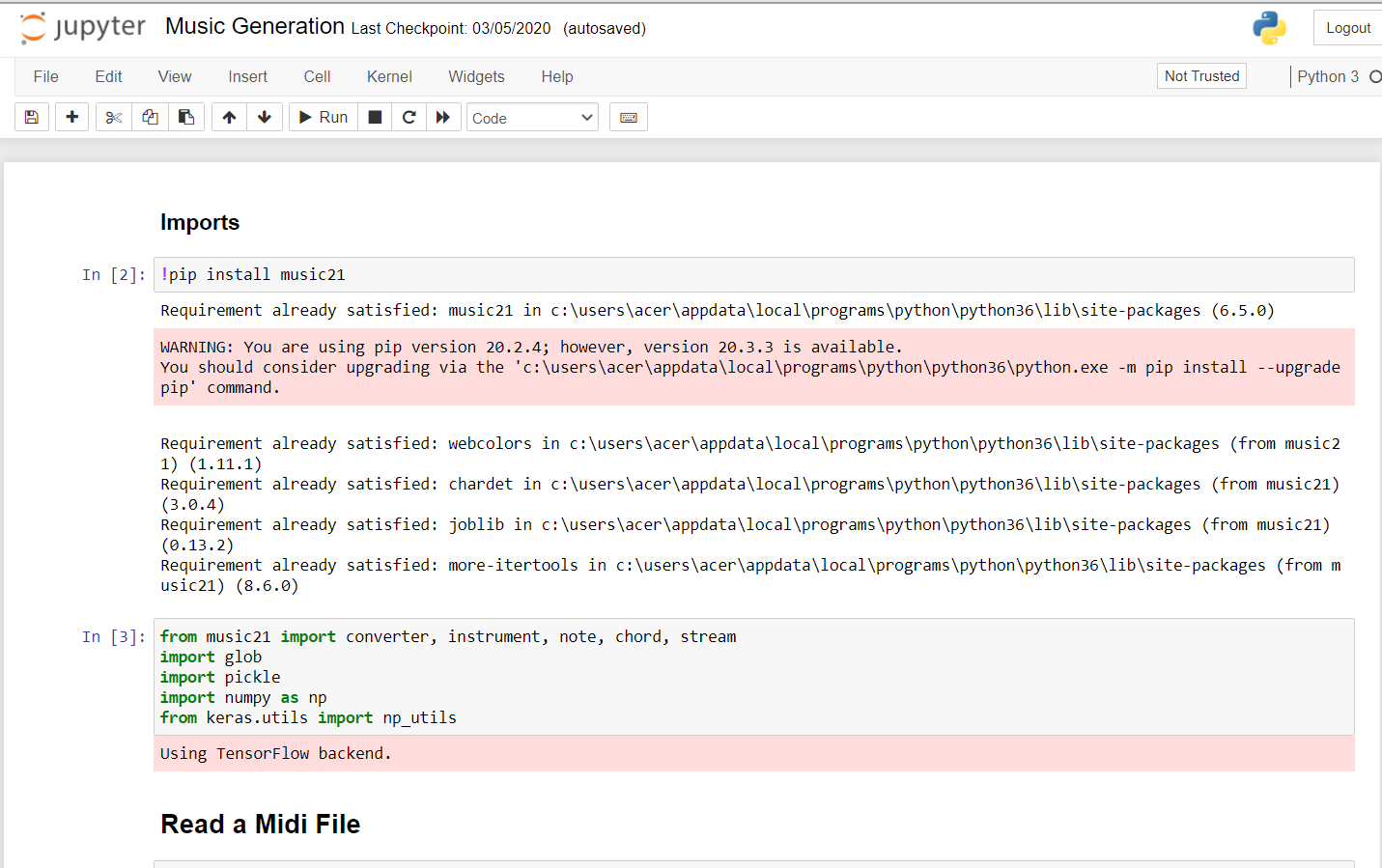
In this chapter the relative study about the project and relative structure of Jarvis was explained. Main modules discovered by me for my project has been discussed a bit and the idea of the project has been highlighted. It would surely give a gist of what is happening in the project.

# Chapter 3

**TOOLS AND TECHNIQUES**

## Software used with technical specifications:

* + - **Python(JUPYTER for programming stuff)**
    - **Various helping, API, speech-recognition and web-scrapping modules**
    - **Other third-party modules**
    1. **Python(JUPYTER):**

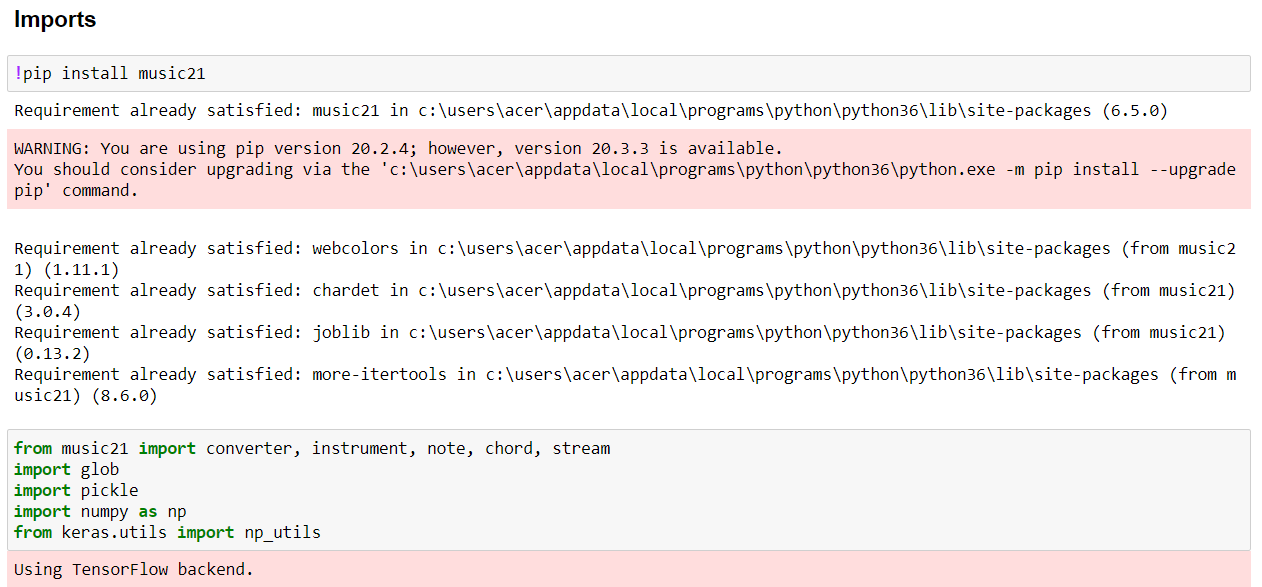
****

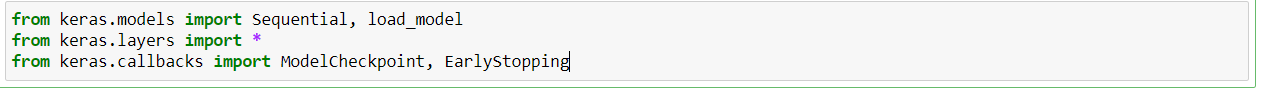
JUPYTER is Python’s Integrated Development and Learning Environment.

JUPYTER has the following features:

* coded in 100% pure Python, using the Keras and tensorflow backend
* cross-platform: works mostly the same on Windows, Unix, and macOS
* Python shell window (interactive interpreter) with colorizing of code input, output, and error messages
* multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features
* search within any window, replace within editor windows, and search through multiple files (grep)
* debugger with persistent breakpoints, stepping, and viewing of global and local namespaces
* configuration, browsers, and other dialogs

## Various python modules involved:

****

****

* + 1. **Main Modules involved:**

Python is a suitable language for script writers and developers. It contains various

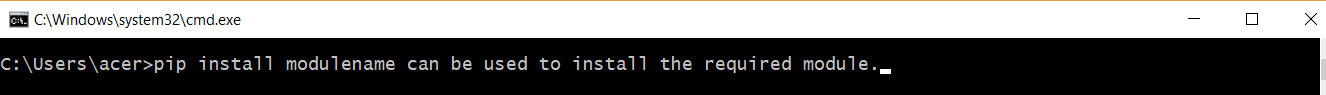
Data science modules like pandas, numpy, keras, tensorflow, scipy, etc which is very much benefitial for a data scientist. This gives and edge to python compared to all other languages which do provide modules but are not easy to use. Here sequential RNN model which is provided by Keras is really useful.

* **Subprocess:-** This module is used for getting system subprocess details which are used in various commands i.e Shutdown, Sleep, etc. This module comes buit-in with Python.
* **Music21:-** It is one of the python modules which can parse midi audio files and can edit them. All the preprocessing part of the project has been done using this module itself.
* **Pickle:-** This module in the project has been used to dump the required binary file so as to use it immediately whenever needed.
* **glob:-** This module basically helps to extract the file with the given extension. It can do so with the help of recursive regular expression matching.
* **numpy:-** Using the numpy library you can get various statistical values in Python. NumPy (Numerical Python) is a module consisting of multidimensional array objects and a collection of routines for processing those arrays. It is in one of those very useful data science modules of python.

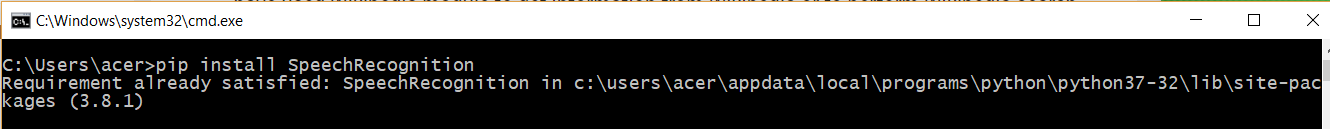
**Data preprocessing:-** Since I have created a data science capstone project, Data extraction, exploration and its cleaning is of utmost importance which I obtained

from Kaggle datasets. Data preprocessing I have done using google colab by switching working mode to GPU which helped me to get my model trained really quickly.

* **Keras:-** This library helps us to have a better deep learning insights which help to visualize and better understand our neural network model. This module does not comes buit-in with Python ,It has to be installed externally.
* **Datetime:-** Date and Time is used to showing Date and Time. This module comes built-int with Python.



**FOR EXAMPLE : -**

****

# “IF THE MODULE IS PREINSTALLED IT WILL SHOW UP THE REQUIREMENT ALREADY SATISFIED”.

# pip:

pip is useful software for installing the third party software in your windows PC.pip is de facto standard package-management system used to install and manage software packaes written in Python.[[4]](https://en.wikipedia.org/wiki/Pip_(package_manager)#cite_note-rhos-pip-4) Many packages can be found in the default source for packages and their dependencies Python Package Index (PyPI).

One major advantage of pip is the ease of its command-line interface, which makes installing Python software packages as easy as issuing a command:

pip install some-package-name

Users can also easily remove the package:

pip uninstall some-package-name

# Summary:

The details of tools used in this project are explained in this chapter. The tools are able to make this project perfect and the modules are also given in this chapter that were used in project. These tools help for efficient output of the program. The JUPYTER output shell can be used to see the output of the program. The program has not been converted to a .exe file and hence has to be run from the console itself. Extra GUI can be added as per the necessity.

# Chapter 4 METHODOLOGIES

## Design of the investigation/Algorithms/ Software:

Basically there is no input command. Just train the required model with various instrument audio in order to get a melodious tone:-

1. Imports are made in the jupyter notebook ide.

2. Read Midi file.

3. Preprocess files.

4. Prepare Sequential Data for LSTM.

5. Create the model

6. Predict and use the predicted notes to generate the required midi file to be stored in on the hard disk.

Hence I can summarize my project overall a combination of ML + DL + modules matter to carry out the work and make it look like the work has been automated by the model and we get the new melodious tone.

## Analysis procedures:

Hard and difficult process of the project is analysis of the project after making it a working software. As it is a simple python project with cross platform dependencies it will work on almost every provided operating system and CPU given the platform has the required modules for proper working of the software.

# Implementation procedure and features including mode of operations:

Here goes the implementation part of the program: -

* 1. First of all collect all the data required for preprocessing from kaggle.
  2. Load all the required midi files and convert songs lyrics to notes and chords and turn everything into flatten list.
  3. Try to parse the notes and chords using music21 module.
  4. As the DL model does not accepts stringified input create a dictionary to map each note with an integer, normalize each value to bring into range of 0-1.
  5. There are 369 distinct class of notes present in our model.
  6. One hot encode the classes as vector and convert it into 3d numpy array with 1 channel. RNN (LSTM cells) model is going to accept a 3d input else you will get an error.
  7. Get the predictions and take argmax of the probabilities generated, after that convert integer mapping to string notes using reverse dictionary mapping.
  8. Now you will be having a list of notes and chords as prediction.
  9. Fix an offset and keep on moving the list predicted data with a stride of 1 and offset 0.5(adjustable according to the music needed).
  10. Get the output list.
  11. You can hear the melody generated in jupyter notebook itself using music21 module.
  12. Save it on the hard disk.
  13. Hurray enjoy with the random audio mixup.
  14. All the elements will be played with same duration of time as the offset is constant. It can be adjusted as per the requirement.

# Disadvantages of the output obtained:

* 1. All files are combined into one note hence our model does not knows how a song should start and how it should end.
  2. It does not knows how soft or harsh the melody should be. It can be adjusted as per the requirement.
  3. Variable timestamps (offset) may lead to some melodious tone.

Changes:

1. We can create some other model with more than one container like guitar + piano + brass ,etc.
2. If we train for more epochs(about 100-200) the model is going to perform surely better.

# Details about hardware:

Having a latest system with nice specs would be a good platform for the software but it may run on almost every system independent of the operating systems used. Given that the system should have the required modules for proper working of the software.

# Summary:

In this chapter methodology of the project was explained. Complete analysis of modules and features of software that were used for this project and output analysis of the project based on the platform dependencies. This chapter was about the analysis and the implementation of the project from the software to hardware and its complete methodology and its functionalities, verification and simulation and complete investigation.

# Chapter 5 RESULTS AND ANALYSIS

This chapter was explained the following result and the analysis of the circuit.

* Software Results
* Parsing and data exploration results.
* modules Results.

## Presentation of the findings:

* + 1. **Hardware results:**
       - For the nice user interface and smooth flow of control there should be a nice internet connection and third party modules installed.
       - Software is platform independent therefore it will run independent of the O.S. used.
    2. **Software results:**

As in the previous chapter explained about the software used in the making of this project there may be small technical glitch while interacting with the midi file directory as it may not be same for the other local pc’s on which my project has to be run.

# Discussion of the findings:

Finding the result obtained from the program by analysis and try to find the error if it’s done.

I found out various errors and how to resolve them like- a 3d tensor has to be fed to the LSTM RNN model and I was trying to feed a linear numpy array. It was a stupid mistake. It was difficult to work with keras as downloading and installing it was also a bit complicated and model compliation would have taken a lot of time if I would not have taken help of Google colab and its GPU processing feature.

**Comparison with initial GOAL:**

Initially I thought of just making it a simple audio mixer and which uses RNN to classify lyrics and generate efficiently but at the end I got a gist that the same model can be used with various other instruments lyrics generation. With the help of Udemy instructor and Harry sir’s YouTube channel I got a lot of knowledge of automating my project and at the end I made far more features which were just a thought to me couple of years ago. Regeneration of lyrics will result in taking a new random state which means a new type of song.

## Recommendations:

* + - * You may add some more features by learning new modules of Python.
      * Do have a look on the source code which would give you an inside depth of how the flow of control goes about.
      * At last but not the least have fun with your own lyrics generator.

## 

## 5.5 Summary:

This chapter is about the results and analysis of the program as the main overcome of the program and its output on the end user’s screen and also telling about the initial goal, comparison of hardware and software results and limitation and recommendation.

# Chapter 6

**CONCLUSION**

Lyrics generation capstone project is upto the expectation which I thought of ever. The main problem that was faced in this project was about searching the required libraries and installing them into my local machine.With very hard work and right online mentorship on this project I am able to make it upto the point. Despite not being a member of devops I have done a lot of trials of the project in adding the source code and modifying stuffs to make it work properly. My udemy instructor is awesome who helped me and gave a drift to the project. Main problem is limitation of data but it can be overcome after having a lot of data and hence overfitting can be avoided and hence we will be able to get more general results. Hence, I would like to conclude it was a good project to maintain the understanding of the python modules and a good read of Data Science (A-Z) with python as well. I enjoyed a lot making this project. I hope the end user of the project would also enjoy it. Features of Lyrics generator are not limited to the ones I have created you may also add a feature to it and enjoy, I have attached my github profile link to make you reach to my source code , Happy coding :) .

# REFERENCES

1. <https://www.codewithharry.com/>
2. [www.google.com](http://www.google.com/)
3. <https://www.geeksforgeeks.org/python-programming-language/>
4. <https://keras.io/api/>
5. https://www.kaggle.com/soumikrakshit/classical-music-midi
6. For source code of the program you may refer to my github profile and contriute it :- <https://github.com/Lucifer7355/JARVIS>