

Instituto Tecnológico de Estudios Superiores de Monterrey

Campus León



Bitácora:

Materia: Matemáticas y ciencia de datos para la toma de decisiones

Integrantes:

A00573517_ Alondra Abigail Torres González

A00573743_ Alejandro Castro Espinosa

A00572455_ Montserrat Rivera González

A00572321_ Emilio Javier Arguelles Alvarado

Fecha: 19 de mayo de 2022

Identifying the genre of a song with neural networks

Introducción:

En el presente proyecto se abordará el tema de ciencia de datos aplicada a las redes neuronales y lo entenderemos por medio del procesamiento de canciones y sonidos para identificar el género musical al que pertenece la melodía. Así mismo, se analizará a detalle desde la teoría para posteriormente, con ayuda del uso de tecnologías y herramientas como Python, resolver este tipo de problemáticas.

El proyecto para trabajar se llama; Identifying the genre of a song with neural networks, y este representará la ejecución de las redes para identificar patrones en las ondas musicales y posteriormente facilitar identificar estas piezas.

Objetivo:

Con este proyecto se pretende lograr aplicar los conocimientos adquiridos a lo largo de la materia de ciencia de datos aplicado a un contexto aleatorio y sencillo que nos llevará a crear curiosidad en este campo para el desarrollo de nuestra carrera.

Marco teórico

- **Neuronal networks (redes neuronales):** estas redes buscan imitar el funcionamiento de las neuronas en un organismo, extraen un modelo de unidades conectadas entre sí y refuerzan conceptos para llegar a conclusiones que se generaran como conocimiento. Estas redes solo trasladan el funcionamiento biológico de las matemáticas a nuestro idioma averiguando los patrones dentro de ellas.
- **Estimulación de canciones según su género:** es un conjunto de datos de mil pistas de audio de 30 seg cada una con diez géneros donde cada uno tiene 100 pistas.

Los géneros son:

- blues
- clásico
- país
- disco
- hip hop

- jazz
 - metal
 - música pop
 - reggae
 - rock
- **Mel-frequency cepstral coefficients (MFCC):** son los coeficientes de la representación del habla basadas en la percepción humana. Se suelen identificar como aplicaciones de reconocimiento de voz.
 - **Music Genre Recognition:** es un campo de la investigación encargado de recuperar la música y sus sonidos, por medio de espectrogramas y la ayuda del MFCC para poder hacer predicciones de los géneros musicales. Es un campo importante de investigación en la

Bitácora

16/05/2022

El primer día discutimos las alternativas de proyectos futuros a trabajar y elegimos el de redes neuronales aplicadas a la música por su complejidad. Sometimos a votación tres posibles opciones de las cuales resultó ganadora la anteriormente mencionada, además nos parece sumamente interesante cómo gracias a esto funcionan algunas aplicaciones como Shazam o SoundHound. Como equipo acordamos trabajar en el proyecto los sábados y domingos por la disponibilidad de todos.

Entre los días 17 y 18 comenzamos a leer el proyecto y a investigar aspectos de utilidad para su desarrollo.

Y el día 19 daremos principio a la elaboración del código en el cual trabajaremos.

19/05/2022

En el siguiente link pudimos utilizarlo de guía para los primeros pasos del código.

<https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners/blob/master/Chapter04/GenreIdentifier.ipynb>

Primer error: No se encuentra ni la librería Librosa ni la librería keras

```

In [11]: import librosa
...: import librosa.feature
...: import librosa.display
...: import glob
...: import numpy as np
...: import matplotlib.pyplot as plt
...: from keras.models import Sequential
...: from keras.layers import Dense, Activation
...: from keras.utils.np_utils import to_categorical
Traceback (most recent call last):

  File "C:\Users\argue\AppData\Local\Temp\ipykernel_17040\814543229.py", line 1, in <module>
    import librosa
ModuleNotFoundError: No module named 'librosa'

```

<https://freesound.org/people/Stereo%20Surgeon/sounds/266093/>

<https://freesound.org/people/grrlrigher/sounds/98195/>

22/05/2022

Tras investigar en foros relacionados con el proyecto notamos qué existe un error en la base datos GTZAN, al parecer el documento .csv no brinda la información necesaria para generar una comparación entre los diferentes géneros de música. Otro problema que encontramos es que al momento de entrar al link que se ubica en el libro no fue posible visualizar el contenido dentro de esta, aparentemente la página no se encuentra.

Link: http://marsyasweb.appspot.com/download/data_sets/.GTZANGenreCollection



Base de datos con 30 segundos:

<https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification>

GTZAN Dataset - Music Genre Classification

Data Code (78) Discussion (11) Metadata 454 New Notebook Download (1 GB)

Music Feature Engineering

Data (2 directories, 2 files)

About this directory

The Data Folder contains:

- genres original folder
- images original folder
- features30seconds.csv file
- features3seconds.csv file

Item	Size
genres_original	10 directories
images_original	10 directories
features_30_sec.csv	1.11 MB
features_3_sec.csv	11.08 MB

Data Explorer
Version 1 (1.41 GB)

- ▾ Data
 - genres_original
 - images_original
 - ▮ features_30_sec.csv
 - ▮ features_3_sec.csv

Summary

- 2001 files
- 120 columns

Patente: https://webhome.cs.uvic.ca/~gtzan/work/pubs/moodlogic_patent_gtzan.pdf

Documento explicando cómo MARSYAS puede ser utilizado para recuperar información musical.

https://www.researchgate.net/publication/228373711_MARSYAS-02_A_case_study_in_implementing_music_information_retrieval_systems

Tratamos de investigar cómo instalar tensorflow y keras en la página oficial de anaconda sin embargo, no funcionó del todo: <https://anaconda.org/conda-forge/tensorflow>

<https://anaconda.org/conda-forge/keras>

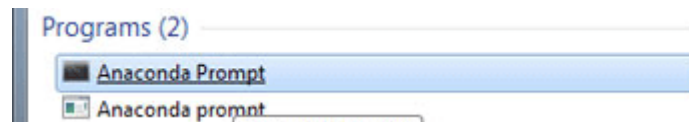
Para tratar de instalar librosa, tensorflow y keras extraídos de las páginas anteriores utilizamos los siguientes comandos:

- `conda install -c conda-forge librosa` **para instalar Librosa (Paquete de Python para análisis de audio y música)**
- `conda install -c conda-forge keras` **para instalar keras (Se usa para ir construyendo por bloques la arquitectura de cada red neuronal, incluyendo redes convolucionales y recurrentes, que son las que permiten, junto a los bloques “más tradicionales”, entrenar modelos deep learning.)**
- `conda install -c conda-forge tensorflow`

Después de un poco de pruebas logramos identificar que debíamos descargar los programas Tensorflow que ayudaría a correr Keras y también logramos descargar Librosa en anaconda.

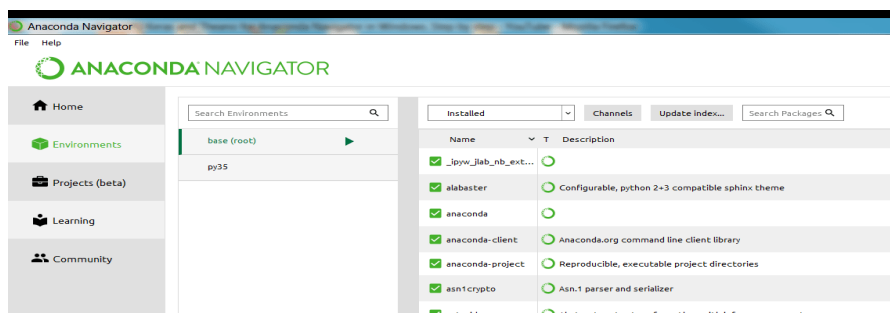
En los intentos por instalar tensorflow en anaconda se hizo el siguiente proceso:

1. Se tuvo que crear una nueva variable del sistema, ejecutando como administrador el **prompt de Anaconda**.



El **prompt de Anaconda** nunca se llegó a cerrar debido a que todo se fue instalando en esa ventana a través de comandos.

- Se escribió y se ejecutó el siguiente comando para crear la nueva variable: “conda create -n py35 python=3.5 anaconda”
- Para después ejecutar la activación: “conda activate py35”
- Nos dirigimos a **Anaconda Navigator** para corroborar que ya estaba instalado. Viendo que en *Enviroments* se puede ver el **py35**. Habiendo ya creado y activado nuestro nueva conda enviroment.



- Después instalamos spyder en nuestra nuevo enviroment con el comando: “conda install spyder”
- A continuación, se nos pidió actualizar conda, aplicando el siguiente comando: “conda update -n py35 anaconda”

- Como siguientes pasos instalamos tensorflow “conda install tensorflow” y al mismo tiempo instalamos tensorflow para CPU :
pip install --ignore-installed --upgrade

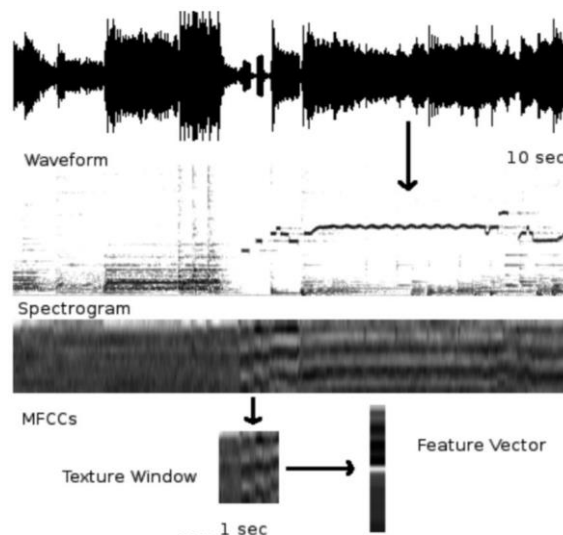
https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_gpu-1.0.1-cp35-cp35m-win_amd64.whl

Dicho proceso, no funcionó.

26/05/2022

Obtuvimos la base de datos de las canciones y seguimos con imprevistos debido al tensor flow

La aplicación de MARSYAS extrae las características del audio para posteriormente traducirlas a un código el cual es la base de muchos algoritmos de Recuperación de información musical (por sus siglas en inglés MIR) que implementan la clasificación, segmentación y la recuperación de similitudes. La siguiente imagen muestra un diagrama de audio el cual extrae las características de este por medio de la segmentación y la clasificación.



Seguimos teniendo problemas con el tensorflow pero seguimos probando los siguientes comandos y posteriormente aparecía en pantalla lo de la siguiente imagen.

To install this package with conda run one of the following:

```
conda install -c conda-forge tensorflow
conda install -c conda-forge/label/broken tensorflow
conda install -c conda-forge/label/cf201901 tensorflow
conda install -c conda-forge/label/cf202003 tensorflow
```

<https://anaconda.org/conda-forge/tensorflow>

[illegible]

Incluso páginas en internet nos arrojaban los mismos comandos, pero seguía deteniéndose a mitad del proceso.

4. To install the current release of CPU-only TensorFlow, recommended for beginners:

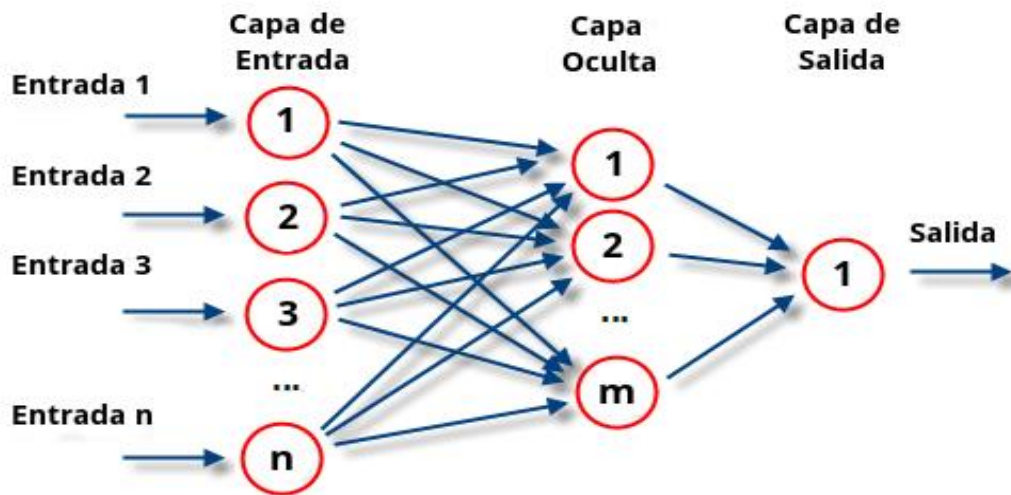
```
conda create -n tf tensorflow
conda activate tf
```

Or, to install the current release of GPU TensorFlow on Linux or Windows:

```
conda create -n tf-gpu tensorflow-gpu
conda activate tf-gpu
```

29/05/2022

¿CÓMO FUNCIONAN LAS REDES NEURALES?

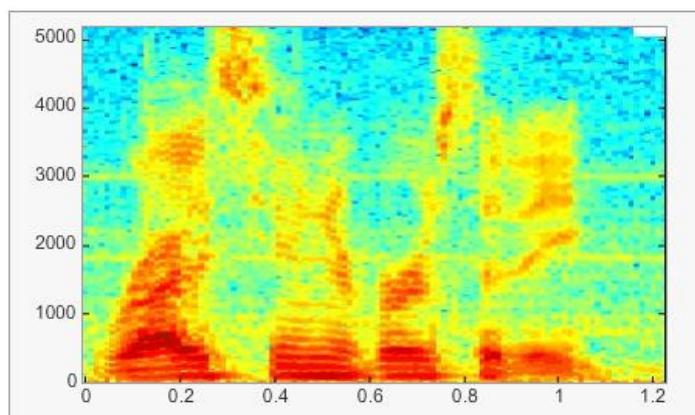


Las neuronas están conectadas unas con otras y estas reciben y mandan impulsos al cuerpo o en este caso a la red computacional. Cuando una neurona obtiene suficiente información de sus vecinos, rápidamente impulsa y manda señales bajo sus axiomas a cada una de las neuronas conectadas.

El humano posee más de 100 billones de neuronas y más de cien trillones de conexiones.

La mayoría de las conexiones neuronales en su mayoría están conectadas, esto quiere decir que cada neurona se conecta a la siguiente capa. Cada neurona de manera individual une valores que recibe en su capa principal, cada conexión de una neurona a la siguiente tiene un peso. Cada neurona tiene una entrada extra llamada "bias" la cual no está conectada a algunas neuronas. Una vez que está cargadas de información una activación es aplicada a la suma.

¿Qué es un espectrograma y sus colores?



La intensidad demuestra la evolución de la frecuencia y de la intensidad en el tiempo, esta se define por el color en cuanto más se aproxima a los colores intensos.

¿Qué es el formato de audio AU?

Es un formato de archivo de audio simple, el cual fue creado por Sun Microsystems. Los cuales se llegan a almacenar los datos en 3 partes, siendo:

- Un encabezado/cabecera
- Cuadra de la anotación de longitud variable o fragmento de información personal
- Datos/Datos de audios reales

Los archivos son grabaciones de audio sin comprimir en un formato diferente y son conocidos como archivos de audio o sonoros que se llegan a utilizar para reproducir o insertar archivos de audio en las páginas Web.

Citas y referencias bibliográficas:

<https://protecciondatos-lopd.com/empresas/tipos-formato-audio/#AU>

<https://www.reviversoft.com/es/file-extensions/au>

01/06/2022

¿Qué tipos de formatos de audios existen?

- WAV y AIFF: son formatos sin compresión, refiriéndose así a copias exactas de la fuente original del sonido, teniendo la misma calidad pero almacenándolos de manera distinta, a AIFF lo creó la compañía de apple a diferencia de WAV siendo este más universal y compatible, su gran desventaja es que utilizan demasiado espacio.

- **FLAC:** es un codec sin pérdida, el más usado del mercado , se usa para no perder calidad usando la compresión por consiguiente, menos espacio, manteniendo el sonido original de la misma calidad siendo mejor que las dos anteriores, además de que su software es gratuito y libre.
- **Apple sin pérdidas (Apple Lossless):** conocido como ALAC, usa compresión y fue elaborado por la compañía de apple pero no es igual de eficiente que la anterior pero es en su mayoría mayormente compatible con iTunes y iOS.
- **APE:** ayuda a ahorra espacio y con una alta calidad similar a los que no tienen perdidas, pero no es compatible en su mayoría de los reproductores por lo que no es muy recomendable por su espacio y compatibilidad.
- **MP3:** MPEG Audio Layer III, es un formato de perdidas, no es el más eficiente, pero si el más compatible a nivel mundial
- **AAC:** Advanced Audio Coding, es muy similar al mp3 pero más eficiente ya que sus archivos ocupan menos espacio y tienen la misma calidad que el formato mp3.
- **Ogg Vorbis:** utiliza el contenedor Ogg, es la versión software libre a MP3 y AAC. No es tan restringido por patentes es similar en cuanto a calidad del mp3 pero muchos reproductores no lo detectan por su poca popularidad.
- **WMA:** Windows Media Audio. Microsoft es propietario de este y no tiene alguna mejora relevante es similar a los demás ya mencionados.

Las matrices en Python tienen la función de almacenar datos en forma de números, cadenas, símbolos, etc, esto se presenta mediante una forma rectangular bidimensional el cual almacena los datos dentro de filas y columnas. La matriz de este código es de 1,000 x 25,000 esto lo sabemos gracias a la creación de una carpeta que almacena los 10 géneros de música con características y etiquetas de 1000 entradas en cada una.

06/06/2022

Seguimos tratando de instalar tensorflow con lo que arrojaba la página oficial de anaconda, pero no funcionaba; el entorno marcaba que no estaba instalada a pesar de ya haber pasado por un proceso.

```
In [10]: import librosa
In [11]: import librosa.feature
In [12]: import librosa.display
In [13]: import glob
In [14]: import numpy as np
In [15]: import matplotlib.pyplot as plt
In [16]: from keras.models import Sequential
In [17]: from keras.layers import Dense,
Activation
In [18]: from keras.utils.np_utils import to
categorical
File "/var/folders/4w/
cxlzdw6j6sv5ntsy99h5bh740000gn/T/
ipykernel_83382/340915639.py", line 1
from keras.utils.np_utils import to
categorical
SyntaxError: invalid syntax
In [19]: from keras.utils.np_utils import
to_categorical
```

Posteriormente solicitamos ayuda a una de nuestras compañeras y nos compartió el siguiente link para instalar Tensorflow y continuar con el programa.

<https://youtu.be/gdHCWds7nQA>

- Se utilizó el comando "pip install tensorflow"

```
Python 3.9.7 (default, Sep 16 2021, 08:50:36)
Type "copyright", "credits" or "license()" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

In [1]: pip install tensorflow
Collecting tensorflow
  Downloading tensorflow-2.9.1-cp39-cp39-macosx_10_14_x86_64.whl (228.5 MB)
    | 81.0 MB 38.5 MB/s eta 0:00:04
Collecting protobuf<3.20,>=3.0.2
  Downloading protobuf-3.19.4-cp39-cp39-macosx_10_9_x86_64.whl (961 kB)
    | 961 kB 16.8 MB/s eta 0:00:01
Requirement already satisfied: h5py<=2.9.0 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from tensorflow) (3.2.1)
Collecting opt-einsum<=2.3.2
  Downloading opt_einsum-3.3.0-py3-none-any.whl (65 kB)
    | 65 kB 6.5 MB/s eta 0:00:01
Collecting tensorflow-estimator<2.10.0,>=2.9.0rc0
  Downloading tensorflow_estimator-2.9.0-py2.py3-none-any.whl (438 kB)
    | 438 kB 32.3 MB/s eta 0:00:01
Collecting termcolor<=1.1.0
  Downloading termcolor-1.1.0.tar.gz (3.9 kB)
Requirement already satisfied: numpy<=1.20 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from tensorflow) (1.20.3)
Collecting tensorflow-io-gcs-filesystem<=0.23.1
  Downloading tensorflow_io_gcs_filesystem-0.26.0-cp39-cp39-macosx_10_14_x86_64.whl (1.6 MB)
    | 1.6 MB 20.7 MB/s eta 0:00:01
Requirement already satisfied: wrapt<=1.11.0 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from tensorflow) (1.12.1)
Requirement already satisfied: typing-extensions<=3.6.6 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from tensorflow) (3.10.0.2)
Requirement already satisfied: packaging in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from tensorflow) (21.0)
Collecting libclang<=13.0.0
  Downloading libclang-14.0.1-py2.py3-none-macosx_10_9_x86_64.whl (13.2 MB)
    | 13.2 MB 8.6 MB/s eta 0:00:01
Collecting absl-py<=1.0.0
  Downloading absl_py-1.1.0-py3-none-any.whl (123 kB)
    | 123 kB 38.4 MB/s eta 0:00:01
Collecting tensorboard<2.10,>=2.9
  Downloading tensorboard-2.9.0-py3-none-any.whl (5.8 MB)
    | 5.8 MB 27.4 MB/s eta 0:00:01
Requirement already satisfied: setuptools in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from tensorflow) (58.0.4)
```

- Después para instalar keras “Pip install keras”

```
In [2]: pip install keras
Requirement already satisfied: keras in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(2.9.0)
Note: you may need to restart the kernel to use updated packages.
```

- Finalmente se instaló librosa con “Pip install librosa”

```
In [8]: pip install librosa
Collecting librosa
  Downloading librosa-0.9.1-py3-none-any.whl (213 kB)
    | 213 kB 1.1 MB/s eta 0:00:01
Collecting audioread<=2.1.5
  Downloading audioread-2.1.9.tar.gz (377 kB)
    | 377 kB 9.8 MB/s eta 0:00:01
Requirement already satisfied: numpy<=1.17.0 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from librosa) (1.20.3)
Collecting resampy<=0.2.2
  Downloading resampy-0.2.2.tar.gz (323 kB)
    | 323 kB 15.7 MB/s eta 0:00:01
Collecting soundfile<=0.10.2
  Downloading SoundFile-0.10.3.post1-py2.py3.cp26.cp27.cp32.cp33.cp34.cp35.cp36.pp27.pp32.pp33-none-macosx_10_5_x86_64.macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.whl (613 kB)
    | 613 kB 25.5 MB/s eta 0:00:01
Requirement already satisfied: scikit-learn<=0.19.1 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from librosa) (0.24.2)
Requirement already satisfied: joblib<=0.14 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from librosa) (1.1.0)
Requirement already satisfied: packaging<=20.0 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from librosa) (21.0)
Requirement already satisfied: numba<=0.45.1 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from librosa) (0.54.1)
Collecting pooch<=1.0
  Downloading pooch-1.6.0-py3-none-any.whl (56 kB)
    | 56 kB 7.5 MB/s eta 0:00:01
Requirement already satisfied: scipy<=1.2.0 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from librosa) (1.7.1)
Requirement already satisfied: decorator<=4.0.10 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from librosa) (5.1.0)
Requirement already satisfied: setuptools in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from numba<=0.45.1->librosa) (58.0.4)
Requirement already satisfied: llvmlite<=0.38,>=0.37.0rc1 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from numba<=0.45.1->librosa) (0.37.0)
Requirement already satisfied: pyparsing<=2.0.2 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from packaging<=20.0->librosa) (3.0.4)
Requirement already satisfied: requests<=2.19.0 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from pooch<=1.0->librosa) (2.26.0)
Requirement already satisfied: appdirs<=1.3.0 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from pooch<=1.0->librosa) (1.4.4)
Requirement already satisfied: charset-normalizer<=2.0.0 in /Users/champ/opt/anaconda3/lib/python3.9/site-packages
(from requests<=2.19.0->pooch<=1.0->librosa) (2.0.4)
Requirement already satisfied: certifi<=2017.4.17 in /Users/champ/opt/anaconda3/lib/python3.9/site-
```

Después de tratar en varias computadoras logramos instalar exitosamente las librerías de Librosa y Keras en la computadora de Alex siguiendo las instrucciones

desde el inicio, de igual manera se instaló el tensor Flow sin que ningún inconveniente ocurriera. Ahora toca descargar la librería con los datos que vamos a necesitar más adelante en el proyecto.

09/6/2022

¿Qué es Librosa feature y Librosa.display?

librosa.feature: Extracción y manipulación de características. Esto incluye la extracción de características de bajo nivel, como cronogramas, espectrograma Mel, MFCC y varias otras características espectrales y rítmicas. También se proporcionan métodos de manipulación de funciones, como funciones delta e incrustación de memoria.

Librosa.display: Encargado de visualizar la información mediante la imagen de un espectrograma, cronograma, etc,

El día de hoy Alondra trató de instalar tensorflow de nuevo y marcaba muchos errores e incluso decía el programa que no había sido instalado.

```
...: import librosa.display
...: import glob
...: import numpy as np
...: import matplotlib.pyplot as plt
...: from keras.models import Sequential
...: from keras.layers import Dense, Activation
...: from keras.utils.np_utils import to_categorical
Traceback (most recent call last):

  File "C:\Users\alond\AppData\Local\Temp\ipykernel_8588\814543229.py", line 7, in
<module>
    from keras.models import Sequential

  File "C:\Users\alond\anaconda3\lib\site-packages\keras\__init__.py", line 21, in
<module>
    from tensorflow.python import tf2

ModuleNotFoundError: No module named 'tensorflow'

In [7]: pip install tensorflow.python
Note: you may need to restart the kernel to use updated packages.
ERROR: Could not find a version that satisfies the requirement tensorflow.python (from
versions: none)
ERROR: No matching distribution found for tensorflow.python

In [8]:
```

- Tratamos de seguir avanzando, pero comenzó a marcar errores en kick-loop.wav

```
In [3]: display_mfcc('kick-loop.wav')
C:\Users\argue\anaconda3\lib\site-packages\librosa\util\decorators.py:88: UserWarning: PySoundFile failed. Trying audioread instead.
  return f(*args, **kwargs)
Traceback (most recent call last):

  File ~\anaconda3\lib\site-packages\librosa\core\audio.py:155 in load
    context = sf.SoundFile(path)

  File ~\anaconda3\lib\site-packages\soundfile.py:649 in __init__
    self._file = self._open(file, mode_int, closefd)

  File ~\anaconda3\lib\site-packages\soundfile.py:1187 in _open
    _error_check(_snd.sf_error(file_ptr),

  File ~\anaconda3\lib\site-packages\soundfile.py:1357 in _error_check
    raise RuntimeError(prefix + _ffi.string(err_str).decode('utf-8', 'replace'))

RuntimeError: Error opening 'kick-loop.wav': System error.
```

- Como siguiente paso se importó pandas y los archivos csv que se encontraban en la carpeta donde se descargaron los archivos de música.

```
In [6]: import pandas as pd

In [7]: df=pd.read_csv("features_3_sec.csv")

In [8]: df2=pd.read_csv("features_30_sec.csv")
```

- Continuamos con el código, pero ahora tuvimos problemas con el formato au.

```
In [11]: display_mfcc('genres/disco/disco.00035.au')
C:\Users\argue\anaconda3\lib\site-packages\librosa\util\decorators.py:88: UserWarning: PySoundFile failed. Trying audioread instead.
  return f(*args, **kwargs)
Traceback (most recent call last):

  File ~\anaconda3\lib\site-packages\librosa\core\audio.py:155 in load
    context = sf.SoundFile(path)

  File ~\anaconda3\lib\site-packages\soundfile.py:649 in __init__
    self._file = self._open(file, mode_int, closefd)

  File ~\anaconda3\lib\site-packages\soundfile.py:1187 in _open
    _error_check(_snd.sf_error(file_ptr),

  File ~\anaconda3\lib\site-packages\soundfile.py:1357 in _error_check
    raise RuntimeError(prefix + _ffi.string(err_str).decode('utf-8', 'replace'))

RuntimeError: Error opening 'genres/disco/disco.00035.au': System error.
```

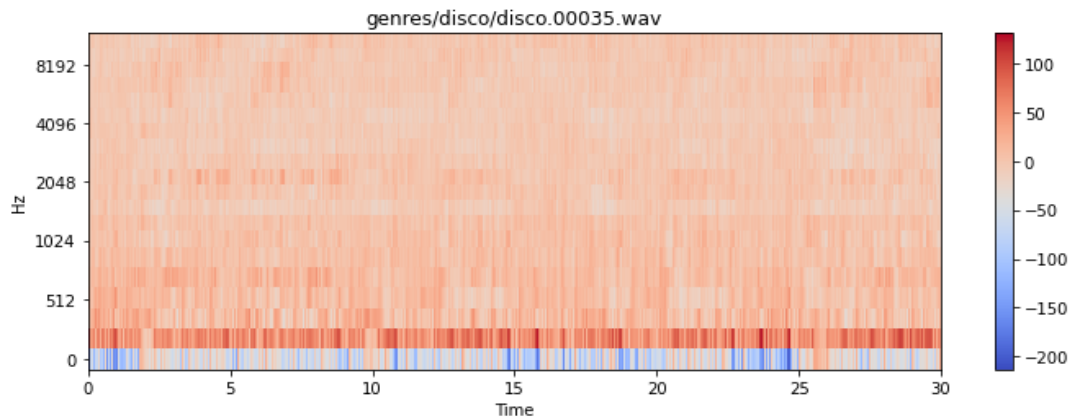
- Para resolver lo anterior simplemente se le cambió la terminación de au a .wav y funcionó.

```
In [12]: display_mfcc('genres/disco/disco.00035.wav')
C:\Users\argue\AppData\Local\Temp\ipykernel_14892\1020391901.py:3: FutureWarning: Pass y=[-0.08514404 -0.08752441 -0.09707642 ... 0.62249756 0.63378906
0.7043152 ] as keyword args. From version 0.10 passing these as positional arguments will result in an error
  mfcc = librosa.feature.mfcc(y)
```

Warning

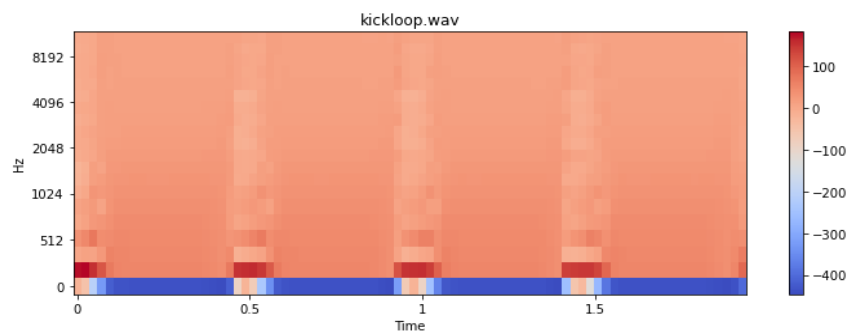
Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.

- Posterior a la solución del anterior error se generó el siguiente espectrograma.



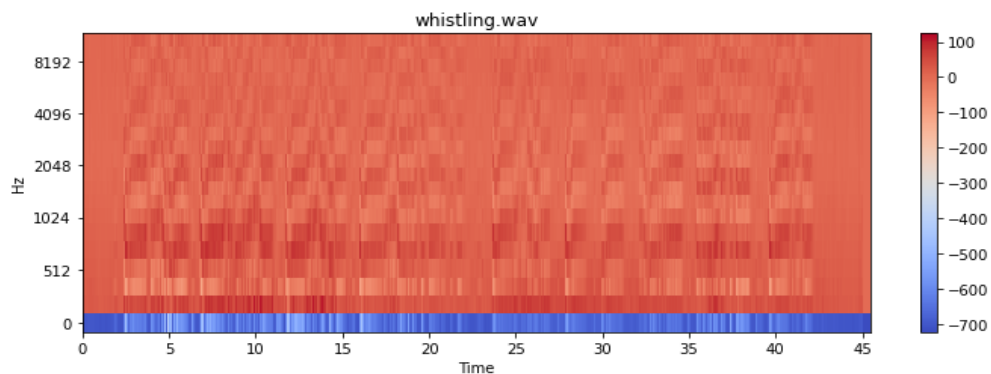
- Después usamos kickloop.wav y formó otro espectrograma como el que se ve a continuación.

```
In [14]: display_mfcc('kickloop.wav')
C:\Users\argue\AppData\Local\Temp\ipykernel_14892\1020391901.py:3: FutureWarning: Pass y=[-1.7467950e-03  2.7376101e-03 -1.0124112e-02 ... -2.3858712e-07
2.0053402e-07  0.0000000e+00] as keyword args. From version 0.10 passing these as positional arguments will result in an error
mfcc = librosa.feature.mfcc(y)
```



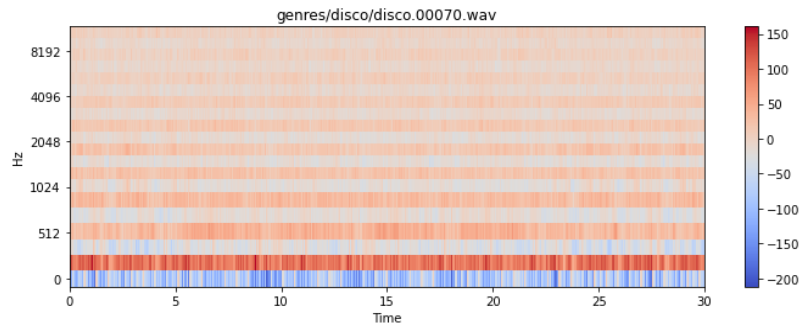
- Se escribió el comando whistling.wav para formar otro espectrograma.

```
In [15]: display_mfcc('whistling.wav')
C:\Users\argue\AppData\Local\Temp\ipykernel_14892\1020391901.py:3: FutureWarning: Pass y=[0.00014207 0.00014386 0.00016641 ... 0. 0. 0.
] as keyword args. From version 0.10 passing these as positional arguments will result in an error
mfcc = librosa.feature.mfcc(y)
```



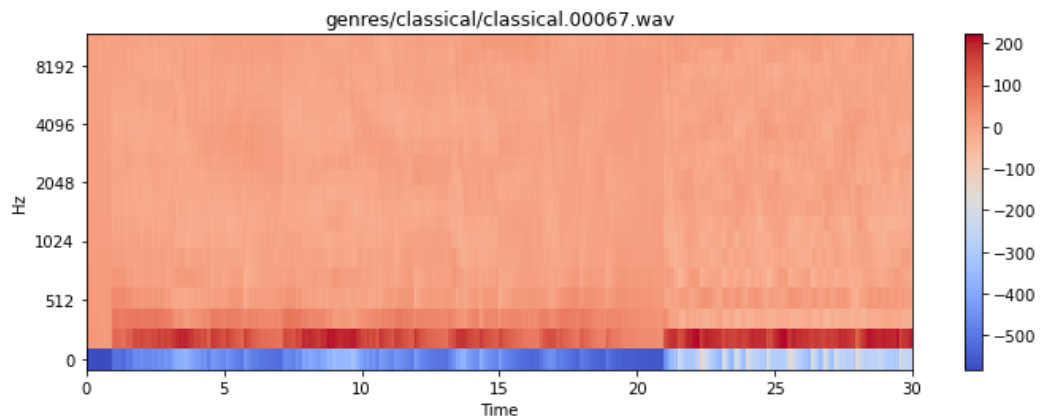
- Se utilizaron de ejemplo dos audios que se encontraban en las páginas de internet anteriormente mencionadas y arrojaron lo siguiente.

```
In [16]: display_mfcc('genres/disco/disco.00070.wav')
C:\Users\argue\AppData\Local\Temp\ipykernel_14892\1020391901.py:3: FutureWarning: Pass y=[-0.06051636 -0.06201172 -0.03417969 ... -0.01367188 -0.06228638
-0.06088257] as keyword args. From version 0.10 passing these as positional arguments will result in an error
mfcc = librosa.feature.mfcc(y)
```

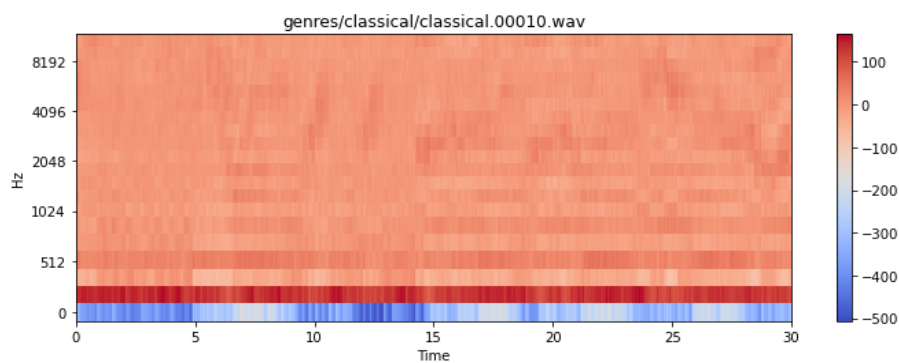


- Display_mfcc

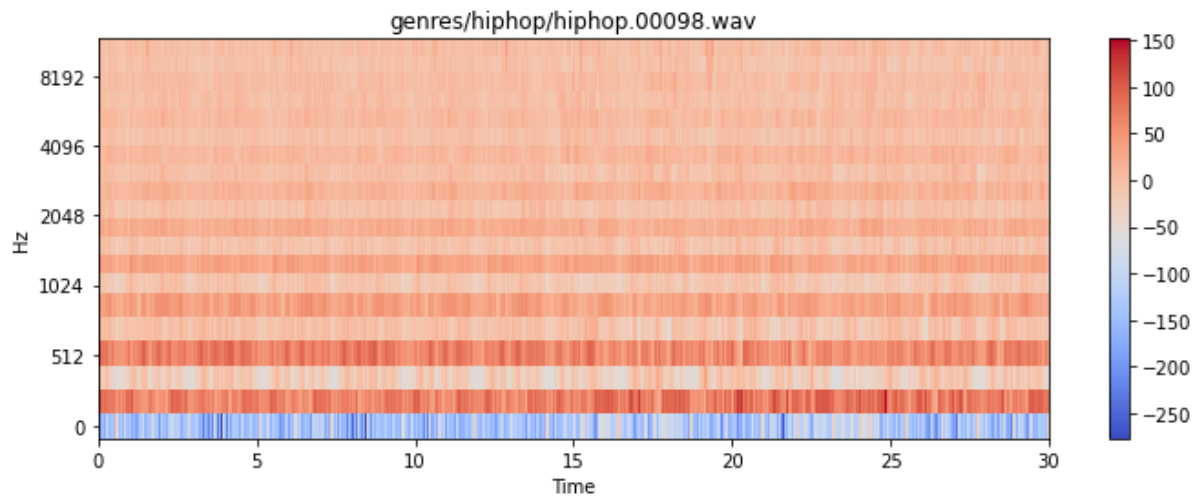
```
In [17]: display_mfcc('genres/classical/classical.00067.wav')
C:\Users\argue\AppData\Local\Temp\ipykernel_14892\1020391901.py:3: FutureWarning: Pass y=[-0.00109863 -0.0010376 -0.00109863 ... -0.04022217 -0.01620483
0.00531006] as keyword args. From version 0.10 passing these as positional arguments will result in an error
mfcc = librosa.feature.mfcc(y)
```



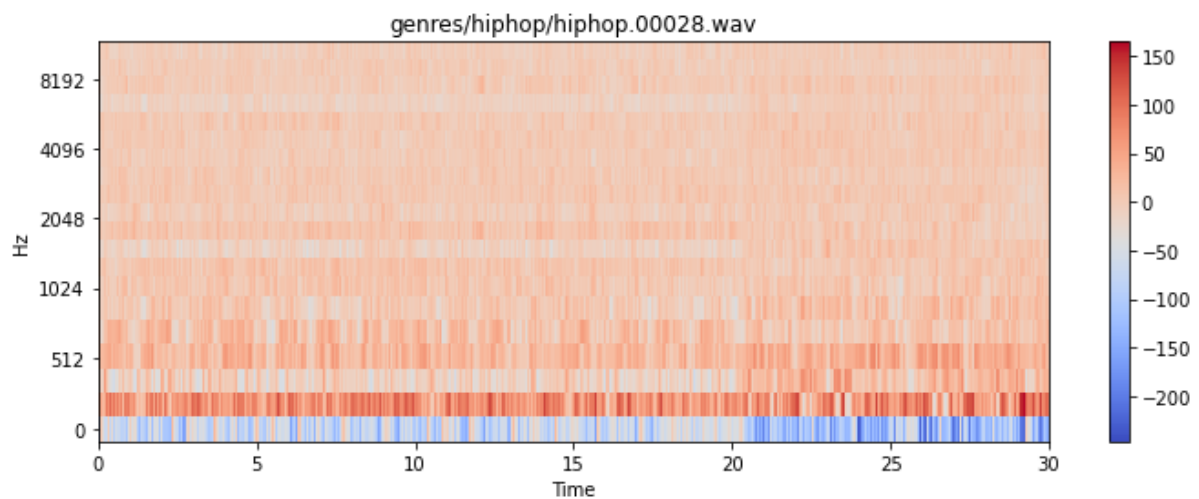
```
In [18]: display_mfcc('genres/classical/classical.00010.wav')
C:\Users\argue\AppData\Local\Temp\ipykernel_14892\1020391901.py:3: FutureWarning: Pass y=[0.01327515 0.02868652 0.03347778 ... 0.03952026 0.05947876
0.06317139] as keyword args. From version 0.10 passing these as positional arguments will result in an error
mfcc = librosa.feature.mfcc(y)
```



```
In [19]: display_mfcc('genres/hiphop/hiphop.00098.wav')
C:\Users\argue\AppData\Local\Temp\ipykernel_14892\1020391901.py:3: FutureWarning: Pass y=[ 0.11871338  0.19989014  0.15310669 ... -0.07220459  0.07394409
 0.1463623 ] as keyword args. From version 0.10 passing these as positional arguments will result in an error
mfcc = librosa.feature.mfcc(y)
```



```
In [20]: display_mfcc('genres/hiphop/hiphop.00028.wav')
C:\Users\argue\AppData\Local\Temp\ipykernel_14892\1020391901.py:3: FutureWarning: Pass y=[-0.00393677  0.02148438  0.04342651 ...  0.06762695  0.05975342
 0.05630493] as keyword args. From version 0.10 passing these as positional arguments will result in an error
mfcc = librosa.feature.mfcc(y)
```



- Continuamos con el comando `def extract_features_song`, utilizando `mfcc` y después devolviendo la orden con `np.ndarray.flatten`

```
In [21]: def extract_features_song(f):
...:     y, _ = librosa.load(f)
...:
...:     # get Mel-frequency cepstral coefficients
...:     mfcc = librosa.feature.mfcc(y)
...:     # normalize values between -1,1 (divide by max)
...:     mfcc /= np.amax(np.absolute(mfcc))
...:
...:     return np.ndarray.flatten(mfcc)[:25000]
```

- Escribimos la siguiente parte del código y no tuvimos ningún problema

```
In [22]: def generate_features_and_labels():
...:     all_features = []
...:     all_labels = []
...:
...:     genres = ['blues', 'classical', 'country', 'disco', 'hiphop', 'jazz', 'metal', 'pop', 'reggae', 'rock']
...:     for genre in genres:
...:         sound_files = glob.glob('genres/'+genre+'/*.au')
...:         print('Processing %d songs in %s genre...' % (len(sound_files), genre))
...:         for f in sound_files:
...:             features = extract_features_song(f)
...:             all_features.append(features)
...:             all_labels.append(genre)
...:
...:     # convert labels to one-hot encoding
...:     label_uniq_ids, label_row_ids = np.unique(all_labels, return_inverse=True)
...:     label_row_ids = label_row_ids.astype(np.int32, copy=False)
...:     onehot_labels = to_categorical(label_row_ids, len(label_uniq_ids))
...:     return np.stack(all_features), onehot_labels
...:
```

- Sin embargo nos encontramos con un último problema, no detecta el archivo de canciones y arroja que hay 0 en las carpetas.

```
In [23]: features, labels = generate_features_and_labels()
Processing 0 songs in blues genre...
Processing 0 songs in classical genre...
Processing 0 songs in country genre...
Processing 0 songs in disco genre...
Processing 0 songs in hiphop genre...
Processing 0 songs in jazz genre...
Processing 0 songs in metal genre...
Processing 0 songs in pop genre...
Processing 0 songs in reggae genre...
Processing 0 songs in rock genre...
Traceback (most recent call last):

  Input In [23] in <cell line: 1>
    features, labels = generate_features_and_labels()

  Input In [22] in generate_features_and_labels
    onehot_labels = to_categorical(label_row_ids, len(label_uniq_ids))

  File ~\anaconda3\lib\site-packages\keras\utils\np_utils.py:68 in to_categorical
    num_classes = np.max(y) + 1

  File <__array_function__ internals>:5 in amax

  File ~\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:2754 in amax
    return _wrapreduction(a, np.maximum, 'max', axis, None, out,

  File ~\anaconda3\lib\site-packages\numpy\core\fromnumeric.py:86 in _wrapreduction
    return ufunc.reduce(obj, axis, dtype, out, **passkwargs)

ValueError: zero-size array to reduction operation maximum which has no identity
```

Interpretación y variables

El programa cumple con su función de identificar el género de las canciones simulando las redes generadas por nuestras neuronas.

Conclusiones

Gracias a la tecnología podemos simular conexiones neuronales gracias a la tecnología y con este programa a pesar de no haberlo concluido se logró generar conciencia de lo que es capaz de hacer las aplicaciones y programas como parte de la vida diaria.

Aprendimos que para que un programa de este tipo funcione se debe adaptar el entorno, los datos, los conceptos; para finalmente ejecutarlo. A pesar de no correr por completo el programa se logró correr en su mayoría y aprender a solucionar los problemas a los que nos enfrentamos.

Evaluaciones:

Aportación de Alejandro Castro:

Monse: Llegó a apoyar de una manera muy importante conforme a la teoría que incluye el proyecto que habíamos elegido, dándole más contenido a toda nuestra bitácora para una mejor comprensión de lo que habíamos elegido y aportó haciendo la presentación del último día.

Alondra: Aportó bastante en la parte de la teoría como en la del código, ya que siempre anduvo investigando en distintas fuentes para que se pudiera lograr correr tensorflow y también para adentrarse más en el tema.

Emilio: Fue el que inició probando las librerías y código en su computadora, por lo cual él fue el que apoyó más en la parte de prueba y error en nuestro problema que habíamos elegido, además de que investigó diferentes fuentes para que se pudiera lograr el objetivo, tal como para correr tensorflow.

Aportación de Emilio:

Monse: Investigó acerca de la teoría que teníamos que desarrollar y conocía todo el contenido necesario. Hizo la presentación final que se presentó el último día.

Alondra: Estuvo investigando y tratando de resolver problemas del código, encontró múltiples métodos para descargar tensorflow y estuvo preguntando a otras personas que ya habían hecho el proyecto.

Alejandro: Además de investigar teoría, también logró conseguir la base de datos actualizará debido a que la que se nos otorgó no estaba completa. Fue el primero en poder correr tensorflow en su computadora.

Aportación de Montserrat:

Durante las 5 semanas el equipo trabajo adecuadamente, Alejandro y Arguelles se encargaron de realizar el código y de resolver los problemas que surgían a medidas que avanzan con este mismo, ya al final lograron concluir el código con ayuda de la computadora de Alejandro. Alondra realizó parte de la teoría de la bitácora y también ayudó a buscar soluciones que se presentaban en el código, uno de ellos fue el tensorflow. Mi participación en el equipo se centró totalmente en la teoría, investigué sobre cómo funcionaba el código y para qué era utilizado en otras plataformas, como la base de datos del proyecto, la cual me llevó a entender el método de trabajo del autor.

Aportación Alondra:

Durante el transcurso de todas las semanas cada uno de los integrantes aportó con lo que pudo, supo e investigó. La computadora de Emilio fue la que más soportó el programa porque las demás se congelaban y por ejemplo a pesar de que no todos podíamos correrlas nos poníamos a investigar, pedir ayuda en foros, preguntar a otros compañeros y cosas similares.

Todos nos desempeñamos considero de una manera eficiente porque a pesar de tener otros problemas y tareas, tratamos de acoplarnos a la demanda del proyecto para sacarlo adelante y simplemente estoy agradecida porque todos aportamos, fuimos creativos y nos explicamos si algo no entendíamos.

Referencias

- Vía | LifeHacker - What's the Difference Between All These Audio Formats, and Which One Should I Use?
- Imagen | Carl Berkeley
- Keras :: Anaconda.org. (s. f.). Anaconda.Org. <https://anaconda.org/conda-forge/keras>
- P. (s. f.-a). *Python-Artificial-Intelligence-Projects-for-Beginners/GenreIdentifier.ipynb at master · PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners.* GitHub. <https://github.com/PacktPublishing/Python-Artificial-Intelligence-Projects-for-Beginners/blob/master/Chapter04/GenreIdentifier.ipynb>
- Please Wait. . . / Cloudflare. (s. f.). ResearchGate Logo. https://www.researchgate.net/publication/228373711_MARSYAS-02_A_case_study_in_implementing_music_information_retrieval_systems
- R. (s. f.-b). *AU Extensión de archivo - ¿Qué es .au y cómo abrir?* - ReviverSoft. ReviverSoft LLC. <https://www.reviversoft.com/es/file-extensions/au>
- *whistling.wav.* (s. f.-a). Freesound. <https://freesound.org/people/grrlrighter/sounds/98195/>
- *whistling.wav.* (s. f.-b). Freesound. <https://freesound.org/people/grrlrighter/sounds/98195/>
- *Mejores tipos de formatos de audio* | Adobe. (s. f.). ADOBE. <https://www.adobe.com/es/creativecloud/video/discover/best-audio-format.html>
- [PY 1] *Install Anaconda, Spyder, Tensor flow and Keras in windows 10/11.* (2021, 11 julio). [Vídeo]. YouTube. <https://www.youtube.com/watch?v=gdHCWds7nQA&feature=youtu.be>
- Figueiras, S., & Figueiras, S. (2022, 4 marzo). *¿Cómo funcionan las Redes Neuronales?* CEUPE. <https://www.ceupe.mx/blog/como-funcionan-las-redes-neuronales.html>

- *gtzan* / *TensorFlow Datasets*. (s. f.). TensorFlow.
<https://www.tensorflow.org/datasets/catalog/gtzan>
- I. (2018, 22 abril). *MFCC implementation and tutorial*. Kaggle.
<https://www.kaggle.com/code/ilyamich/mfcc-implementation-and-tutorial/notebook>