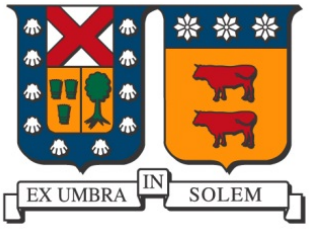
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



“CLASIFICACIÓN MULTI-ETIQUETA DE GÉNEROS MUSICALES DE SPOTIFY A PARTIR DE LA SEPARACIÓN DEL AUDIO EN FUENTES MUSICALES”

YOEL BERANT ELORZA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Marcelo Mendoza
Profesor Correferente: Claudio Lobos

Septiembre - 2022

DEDICATORIA

Dedicado a mi familia: a mis padres, mi hermano, mis tíos y primos.

AGRADECIMIENTOS

Agradezco a mi familia y a mis amigos cercanos por apoyarme en todo este proceso, no solo en esta memoria sino en todos los años que duró mi carrera en esta Universidad. Agradezco a mis profesores por transmitirme la pasión por la informática. Finalmente agradezco a la fundación B'nai B'rith por creer en mi al ayudarme todos estos años con una beca.

RESUMEN

El problema de la clasificación automática de géneros musicales es un tópico que resulta relevante considerando el auge de los servicios de streaming musicales en los últimos años y las herramientas que estos pueden ofrecer, como los sistemas de recomendación. Esta investigación planea abordar este problema desde el enfoque multi-etiqueta, es decir, considerando los casos en los que múltiples géneros musicales se hagan presentes en una canción. Para abordar este desafío, se plantea usar un conjunto de canciones extraídas desde Spotify separando los archivos de audio en cuatro fuentes musicales: vocalizaciones, percusiones, bajos e instrumentalizaciones u otros. Se plantea también utilizar dos enfoques de redes neuronales convolucionales: un enfoque de modelo complejo o "clásico", es decir, una sola red que identifique la presencia de varios géneros musicales; y un enfoque basado en *committe machines* o, en otras palabras, varias redes independientes las cuales cada una tendrá que detectar la presencia de un género musical en específico a partir de la canción. Los resultados demuestran que a medida que los géneros musicales se van ramificando en múltiples subgéneros estos son cada vez más difíciles de reconocer por parte de los modelos de aprendizaje, pues poseen características cada vez más vagas y menos definidas que los distinguen. En cuanto a los dos enfoques de modelos propuestos, el enfoque basado en *committe machines* logra un desempeño inferior al enfoque "clásico", pues para entrenar cada red independiente de forma que logre buenos resultados se necesita un conjunto de datos específico que debe cumplir con dos requisitos difíciles de cumplir dada las condiciones presentadas durante esta investigación: poseer una cantidad grande de canciones y estar balanceado en cuanto a las etiquetas de las canciones.

Palabras Clave— Géneros musicales; redes neuronales convolucionales; procesamiento de audio; subgéneros; modelos de aprendizaje.

ABSTRACT

The automatic music genre classification problem is an interesting topic which is relevant regarding the growing of music streaming services on the last years and the tools that those services can offer, like recommendation systems. This investigation aims to tackle this problem from the multi-tag approach, or in other words, considering the cases in which multiple music genres may appear in a song. To tackle this challenge, using a dataset of

songs extracted from Spotify is proposed, separating the audio data in four music sources: vocals, drums, basses and instrumentalizations or others. Two convolutional neural network approaches are also proposed: a complex or “classic” model approach, in which only one neural network must identify the presence of various music genres in a song and a committee machines approach, in which multiple neural networks will have to predict from the input song the presence of a specific music genre assigned to each neural network. The results show that as music genres branch out into multiple subgenres, these subgenres are harder to identify by the learning models because they have more vague and harder to define features to distinguish them. Regarding the two proposed model approaches, the committee machines approach achieves lower performance than the “classic” approach because in order to achieve good results, each independent neural networks has to be trained using a dataset which satisfies two requirements that are hard to achieve in the conditions presented during this investigation: having a large number of songs and be balanced in terms of their songs tags.

Keywords— Music genres; convolutional neuronal networks; audio preprocessing; subgenres; learning models.

GLOSARIO

ANN: *Artificial Neural Network* o red neuronal artificial.
BR: *Binary Relevance* o relevancia binaria.
CNN: *Convolutional Neural Network* o red neuronal artificial.
DFT: *Discrete Fourier Transform* o transformada discreta de. Fourier
FN: *False Negatives* o falsos negativos
FP: *False Positives* o falsos positivos
IRLBl: *Imbalance Ratio per Label* o ratio de imbalance por etiqueta
LP: *Label Powerset*
STFT: *Short Time Discrete Fourier Transform* o transformada discreta de Fourier a corto plazo.
TN: *True Negatives* o verdaderos negativos
TP: *True Positives* o verdaderos positivos

ÍNDICE DE CONTENIDOS

GLOSARIO VI

ÍNDICE DE FIGURAS IX

ÍNDICE DE TABLAS X

INTRODUCCIÓN 1

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA 2
1.1 CONTEXTO Y SITUACIÓN ACTUAL 2
1.2 CLASIFICACIÓN DE GÉNEROS MUSICALES MULTI-ETIQUETA 3
1.3 OBJETIVOS 4

CAPÍTULO 2: MARCO CONCEPTUAL 6
2.1 PREPROCESAMIENTO DE AUDIO 6
2.1.1 SONIDO 6
2.1.2 SAMPLEO 6
2.1.3 TRANSFORMADA DE FOURIER Y DFT 8
2.1.4 STFT Y FRAME SIZE 9
2.1.5 HOP SIZE 9
2.1.6 ESPECTROGRAMAS 9
2.1.7 ESCALA Y ESPECTROGRAMA DE MEL 10
2.2 MODELOS DE PREDICCIÓN 12
2.2.1 DEFINICIÓN BÁSICA 12
2.2.2 REDES NEURONALES ARTIFICIALES 13
2.2.3 CONJUNTOS DE ENTRENAMIENTO, VALIDACIÓN Y PRUEBAS 14
2.2.4 REDES NEURONALES CONVOLUCIONALES Y MAPAS DE CARACTERÍSTICAS 14
2.3 CLASIFICACIÓN MULTIETIQUETA 16
2.3.1 COMMITTE MACHINES 18
2.4 MÉTRICAS DE EVALUACIÓN 18
2.4.1 CLASIFICACIÓN BINARIA 18
2.4.2 ACCURACY 19
2.4.3 PRECISSION 19
2.4.4 RECALL 19
2.4.5 F1-SCORE 20
2.4.6 MEDIDAS MACRO Y MICRO 20
2.5 MATRIZ DE CONFUSIÓN 22

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN 24
3.1 PLATAFORMA 24
3.2 OBTENCIÓN DE BASE DE DATOS MULTIETIQUETA 24
3.3 OBTENCIÓN DE PISTAS DE AUDIO 35

3.4 SEPARACIÓN DE INSTRUMENTOS . . . 35
3.5 MUESTREO Y REPRESENTACIÓN DE ESPECTROGRAMAS . . . 36
3.6 USO DE REDES NEURONALES CONVOLUCIONALES . . . 38
3.6.1 MODELO COMPLEJO O CLÁSICO . . . 39
3.6.2 COMMITTE MACHINE . . . 41

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN . . . 44
4.1 evaluación de modelos . . . 44
4.1.1 métricas de evaluación . . . 44
4.1.2 resultados . . . 45
4.1.3 Análisis general . . . 47
4.1.4 Comparación de los dos modelos . . . 48
4.1.5 Análisis de los resultados en géneros en específico . . . 49
4.2 Matriz de confusión . . . 52
4.3 visualización de mapas de características . . . 55

CAPÍTULO 5: CONCLUSIONES . . . 66
5.1 Conclusiones . . . 66
5.2 Trabajo a futuro . . . 68

ANEXOS . . . 69

REFERENCIAS BIBLIOGRÁFICAS . . . 70

ÍNDICE DE FIGURAS

1 Resultado de una encuesta que muestra las preferencias de los americanos en relación a métodos para escuchar música en 2017 y ahora (2021). 2

2 Captura de un espacio en el que se visualizan algunos géneros musicales. Los géneros que se muestran se posicionan en el espacio según la cercanía entre ellos, por lo que géneros más cercanos aparecerán juntos. 4

3 Ejemplo de sampleo de una onda de audio periódica. Notar que los puntos se toman en un *sampling rate* mucho mayor a la frecuencia de la onda. 7

4 ejemplo práctico de posibles reconstrucciones (b), (c) y (d) de una onda de sonido a partir de una señal discreta en puntos sampleados en (a). 8

5 Espectrograma de un violín. Al observar el espectrograma, se observa que la mayor parte de las frecuencias que componen al sonido del violín en toda su duración a través del tiempo se encuentran entre los 440 Hz y los 5.1 kHz. . . . 10

6 Escala de mel como una función de la frecuencia 11

7 Ejemplo de un espectrograma de mel. Nótese que entre los 512[Hz] y los 1024[Hz], y entre los 1024[Hz] y los 2048[Hz] las distancias visualmente son las mismas, a pesar de que las diferencias en [Hz] no lo son. 12

8 arquitectura simplificada de una ANN. Consiste en la capa de entrada a la izquierda, las capas ocultas al centro y la salida a la derecha. 13

9 Ejemplo de procesamiento de un kernel en una CNN. En este caso, el kernel es una matriz de tamaño 3×3, cuyos elementos se multiplican por los de un segmento de la entrada de tamaño 3×3 y se suman en la matriz de salida dando un valor de 9. El mismo kernel repetirá el ejercicio para los otros segmentos de tamaño 3×3 de la entrada hasta completar la matriz de salida. 15

10 Ejemplo de aplicación de varias convoluciones a la imagen de un mapache. Notar que en los resultados resaltan patrones como los bordes del mapache, su pelaje o los bordes de las plantas en las que se recuesta. 16

11 Ejemplo de situación multi-etiqueta. 17

12 Ejemplo de relevancia binaria. 17

13 Ejemplo de label powerset. 18

14 Matriz de confusión binaria 22

15 Matriz de confusión multi-clase, en el caso de 3 clases 23

16 Matriz de coocurrencias de géneros en D 28

17 Frecuencias de apariciones de géneros en los conjuntos de canciones antes y después del balanceo de datos. Las frecuencias D_{test} también se muestran. Notar que el balance de apariciones en D_{train} y D_{val} está mucho más equilibrado luego del balanceo de datos. 34

18 A la izquierda, la separación del salvapantallas de windows xp en canales RGB. A la derecha, la separación de un fragmento visualizado en un espectrograma de mel de 15 segundos de la canción *Never Gonna Give you Up* de Rick Astley en cuatro fuentes musicales. 36

19 Diagrama del preprocesamiento de una canción individual en N muestras . . . 38

20 Diagrama de la arquitectura de la red neuronal compleja. Las capas grises representan capas convolucionales, las capas rojizas representan MaxPooling y Dropout y la capa celeste es la capa densa. 40

21 Diagrama de la arquitectura de la red neuronal *committee machine*. 42

22 matriz de confusión para los resultados de la evaluación del modelo complejo en D_{test} 53

23 58

24 59

25 59

26 61

27 62

28 62

29 64

30 64

31 65

ÍNDICE DE TABLAS

1 Géneros musicales seleccionados del conjunto P. 26

2 Resultados de la evaluación del modelo complejo 45

3 Resultados de la evaluación del modelo commette machines 46

INTRODUCCIÓN

En los últimos años los servicios de *streaming* han cambiado radicalmente la forma de escuchar música. Muchos servicios como Spotify, Apple Music o Tidal compiten día a día por la atención de millones de usuarios, desde los más casuales hasta los más melómanos. Para mantenerse a flote, es necesario que los servicios de *streaming* posean factores que los hagan atractivos como una gran librería de canciones, una buena calidad de audio o sistemas de recomendación que den buenos resultados. Todos estos elementos, en especial los sistemas de recomendación, serían difíciles de lograr sin una buena organización de toda la música que se ofrece.

Organizar manualmente las millones de canciones que un buen servicio de *streaming* posee [Pendlebury, 2021] para un sistema de recomendación no es tarea sencilla, por lo que automatizar este proceso suele ser una mejor alternativa. En este escenario sale a flote el reconocimiento automático de géneros musicales de las canciones.

Se proponen dos modelos de aprendizaje automático basados en redes convolucionales para abordar el problema de la clasificación de géneros musicales de canciones en el caso multi-etiqueta, es decir, en el caso de que una canción pueda pertenecer a más de un género. El primer modelo consiste en una red neuronal compleja que intente identificar la presencia o ausencia de varios géneros musicales en una canción de entrada, mientras que el segundo modelo consiste en varias redes neuronales, cada una con la misión de identificar la presencia o ausencia de un género musical en específico en la canción.

La lista de canciones con las que se trabajó fue extraída de la biblioteca de Spotify. En este documento se explica el método de selección de canciones y el balanceo de datos y etiquetas.

En cuanto al preprocesamiento de las canciones utilizadas, se decidió implementar un modelo de aprendizaje pre-entrenado llamado demucs [Défossez *et al.*, 2019], capaz de generar a partir de una canción de entrada una separación de 4 fuentes musicales musicales: la percusión de la canción, la parte vocalizada, los bajos y la instrumentalización u “otros”.

Una vez evaluados los dos modelos de aprendizaje ya entrenados, se visualizarán algunos de los mapas de activación entregados por las capas de las redes neuronales, con el objetivo de encontrar patrones característicos de algunos géneros musicales.

CAPÍTULO 1
DEFINICIÓN DEL PROBLEMA

1.1. CONTEXTO Y SITUACIÓN ACTUAL

Sin lugar a duda el streaming se convirtió en el método predilecto para escuchar música en los últimos años. En la era más próspera del internet, la gente ha preferido usar servicios de streaming musicales como Spotify, Apple Music, Tidal o YouTube Music por sobre otros métodos como la radio, formatos físicos (como los CDs) o directamente descargar música como archivos digitales. Según un estudio hecho por CBS News [Barkus, 2021] el 47 % de americanos prefieren escuchar música mediante servicios de streaming, superando al 31 % que elige a la radio (ver figura 1).

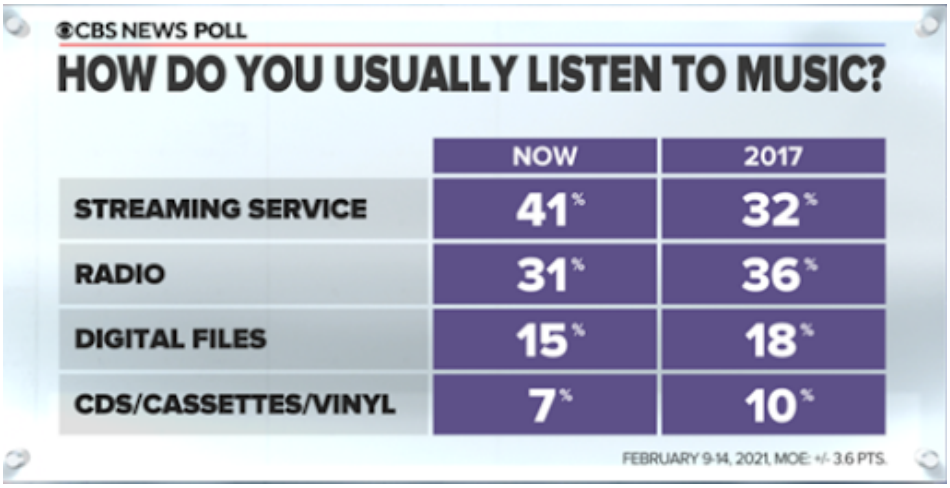


Figura 1: Resultado de una encuesta que muestra las preferencias de los americanos en relación a métodos para escuchar música en 2017 y ahora (2021).
Fuente: CBS News[Barkus, 2021].

Uno de los factores clave detrás del atractivo de los servicios de streaming musicales es el tamaño de las bibliotecas que estos ofrecen. En 2021, Apple Music contaba con más de 90 millones de canciones, Tidal con más de 80 millones, Spotify con más de 70 millones y Amazon Music Unlimited con más de 75 millones de canciones [Pendlebury, 2021].

Por supuesto, poseer una cantidad tan vasta de música conlleva beneficios, pero también desafíos. Los algoritmos de **sistemas de recomendación** y de creación automática de playlists permiten que los usuarios descubran nueva música de acuerdo con sus gustos entre las extensas bibliotecas de canciones que estos servicios ofrecen.

Entre muchas otras cosas, es importante que un sistema de recomendación sea capaz de en-

contrar similitudes entre sus productos [Melville y Sindhwani, 2010]. En el caso de los servicios de streaming musicales esto se traduce en identificar qué canciones, álbumes y artistas se parecen entre sí para que, si un usuario escucha música de un estilo, el sistema le recomiende nueva música de ese mismo estilo. Una de las formas más simples de abordar este problema es **clasificando las canciones según su género musical**.

1.2. CLASIFICACIÓN DE GÉNEROS MUSICALES MULTI-ETIQUETA

El problema de la clasificación automática de géneros musicales ha sido abordado de muchas maneras ([Dai *et al.*, 2016], [Tzanetakis y Cook, 2002], [Li *et al.*, 2003]). Sin embargo, la gran mayoría de acercamientos propuestos han sido utilizados para el caso **multi-clase** del problema, es decir, asumiendo que las canciones a clasificar pertenecen a un y solo un género musical entre muchos. El caso **multi-etiqueta**, en cambio, considera la posibilidad de que una canción pueda incluir múltiples "etiquetas" simultáneamente o, en otras palabras, pertenecer a más de un género musical. El caso multi-etiqueta no ha sido tan explorado como el caso multi-clase [Oramas *et al.*, 2017].

Desde que la tecnología empezó a permitir que la música se grabe [Beardsley y Leech-Wilkinson, 2009], el siglo XX y lo que llevamos del XXI ha sido marcado por el nacimiento y auge de numerosos géneros musicales, cada uno fruto de la experimentación y del anhelo artístico. Algunos de los ejemplos más conocidos son: el Blues, el Jazz, el Rock, el Hip-Hop, lo que hoy conocemos como pop y todas las variantes de los géneros mencionados [thepeoplehistory.com, 2017]. Por su parte, hay que tomar en cuenta que el desarrollo de ninguno de estos géneros es independiente de los demás. Durante la historia de la música "popular" ha existido una tendencia en la experimentación con la mezcla de múltiples estilos musicales [Gandhi *et al.*, 2017]. Por lo tanto, es muy frecuente encontrar en esta historia canciones que pertenezcan a más de un género.

Otra cosa que se debe tomar en cuenta es el alto número de géneros musicales que se han creado hasta la fecha entre tantas variaciones y experimentaciones. Spotify registra más de 5000 géneros [Rodgers, 2020], desde los más comunes como el Rock o el Pop hasta géneros más extraños como "*escape room*", "*noise*" o " *rain*" (que es simplemente sonidos de lluvia). La página web "everynoise.com" [McDonald, 2013], presenta un mapa en el cual se muestran algunos de estos géneros, ubicados en un espacio de dos dimensiones (ver figura 2).

Figura 2: Captura de un espacio en el que se visualizan algunos géneros musicales. Los géneros que se muestran se posicionan en el espacio según la cercanía entre ellos, por lo que géneros más cercanos aparecerán juntos.
Fuente: everynoise.com[McDonald, 2013].

Considerando la altísima cantidad de géneros musicales y la tendencia de estos a mezclarse en canciones, ¿no sería más útil para un sistema de recomendación un modelo de clasificación que pueda distinguir múltiples géneros musicales en una canción en lugar de uno solo?.

Para esta memoria se propone abordar el problema de la clasificación multi-etiqueta de géneros musicales de canciones. Además, considerando la relevancia ya explicada de los servicios de streaming musicales y sus vastas bibliotecas, se plantea utilizar como base de datos canciones extraídas de Spotify, uno de los servicios de streaming más importantes de esta época. Spotify ofrece una lista muy extensa de canciones en las que se presentan diversos géneros musicales.

1.3. OBJETIVOS

Antes de listar a los objetivos se hablará brevemente sobre la propuesta, a nivel técnico. En la sección *marco teórico* se realizará una explicación más detallada de algunos de los conceptos que se mencionan a continuación.

Para abordar el desafío impuesto en la sección anterior se plantea realizar un preprocesamiento de audio utilizando un modelo de aprendizaje pre-entrenado llamado demucs [Défossez *et al.*, 2019], capaz de separar una pista de audio en distintas fuentes musicales, generando 4 nuevas pistas: una con las vocalizaciones de la canción, una con las percusiones, una con los bajos y una última con las instrumentalizaciones u “otros”.

Por supuesto, sería extremadamente difícil desarrollar un modelo que identifique a todos los géneros disponibles en Spotify, por lo que se propone simplificar el problema considerando solo a $N >= 30$ de los géneros más frecuentes en la base de datos obtenida.

Se plantea utilizar y comparar dos modelos de aprendizaje basado en redes neuronales convolucionales. El primero consistirá en una sola red neuronal compleja que identifique la presencia o la ausencia de los N géneros dentro de una canción de entrada y el segundo estará

compuesto de N redes neuronales simples, donde cada red buscará determinar si la canción pertenece o no a uno de los N géneros.

Objetivo General:

Desarrollar y evaluar un modelo de aprendizaje que busque clasificar automáticamente a qué género o géneros musicales pertenece una canción de entrada utilizando una biblioteca de canciones extraída de Spotify, con el fin de evaluar si es posible encontrar similitudes entre canciones (en este caso de un mismo género) a partir del audio preprocesado. De esta forma, en el futuro, se podrá adaptar este modelo de aprendizaje a un sistema de recomendación de música en caso de que el desempeño de la máquina de aprendizaje sea aceptable.

Objetivos Específicos:

- Obtener desde Spotify una base de datos de canciones con sus géneros suficientemente grande y diversa, para así entrenar a un modelo multi-etiqueta que clasifique $N >= 30$ de los géneros más frecuentes.
- Usar una representación de las canciones que permita facilitar el proceso de clasificación por parte de un modelo de aprendizaje. En otras palabras, preprocesar las canciones de una forma correcta, separándolas por instrumentos y pasando las pistas a espectrogramas.
- Evaluar el desempeño del modelo de aprendizaje en cuanto a la clasificación como un todo y en cuanto a los géneros musicales en particular, es decir, determinar a los géneros musicales más fáciles de identificar y a los más difíciles.
- Interpretar los resultados entregados por el clasificador, al visualizar los mapas de activación de las convoluciones, identificando qué partes de la entrada son las que más aportan a la tarea.
- Comparar el desempeño de dos acercamientos basados en redes neuronales: red neuronal compleja (un clasificador de $G >= 30$ etiquetas) y committee machines ($G >= 30$ clasificadores cada uno de una etiqueta).

CAPÍTULO 2
MARCO CONCEPTUAL

2.1. PREPROCESAMIENTO DE AUDIO

2.1.1. SONIDO

El sonido es una onda mecánica resultante de una perturbación en un medio elástico (como el aire) y que se propaga hasta llegar a nuestros tímpanos, generando que estos vibren y se perciban a través del sentido de la audición.

Existen varias características de una onda de sonido que distinguen el cómo se oye, como la frecuencia y la amplitud. La **amplitud** de una onda de sonido, típicamente medida en decibeles([Db]), indica la intensidad en la que se percibe. La **frecuencia** de una onda de sonido, que se suele medir en hercios (1[Hz] = 1/1[S]) determina el tono del sonido; un sonido agudo tiene alta frecuencia mientras que uno grave tiene baja frecuencia.

2.1.2. SAMPLEO

El sonido es una onda continua. Esto implica que una computadora no es capaz de procesar la totalidad de un sonido, puesto que tendría que almacenar infinitos valores de amplitud en un rango de tiempo (dado que son infinitos instantes en un rango de tiempo). Es por esto que si se pretende guardar datos de un sonido en una computadora es necesario transformar la señal sonora a un formato discreto.

Este proceso se conoce como **sampleo**, y consiste en tomar de una onda de sonido varios puntos o "*samples*" (como en la figura 3) según un *"sample rate"*, que dictamina la frecuencia con respecto al tiempo en la que se "marca" un punto de la onda. El *sample rate*, al igual que la frecuencia del sonido, se mide en hercios y su valor estándar es de 44.1[kHz] para audio en formato CD [Velardo, 2020].

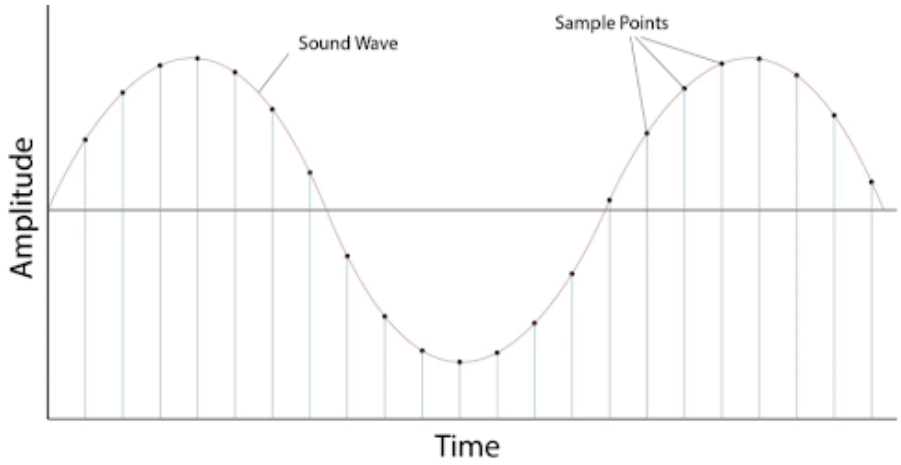


Figura 3: Ejemplo de sampleo de una onda de audio periódica. Notar que los puntos se toman en un *sampling rate* mucho mayor a la frecuencia de la onda.
Fuente: [Velardo, 2020].

Una vez tomados los puntos, es posible reconstruir con cierta precisión la onda de sonido original usando un proceso de interpolación. Si el *sample rate* es lo suficientemente grande, se puede reconstruir un sonido que será percibido por el oído humano como indistinguible al sonido original. La figura 4 ejemplifica posibles reconstrucciones de una manera sencilla.

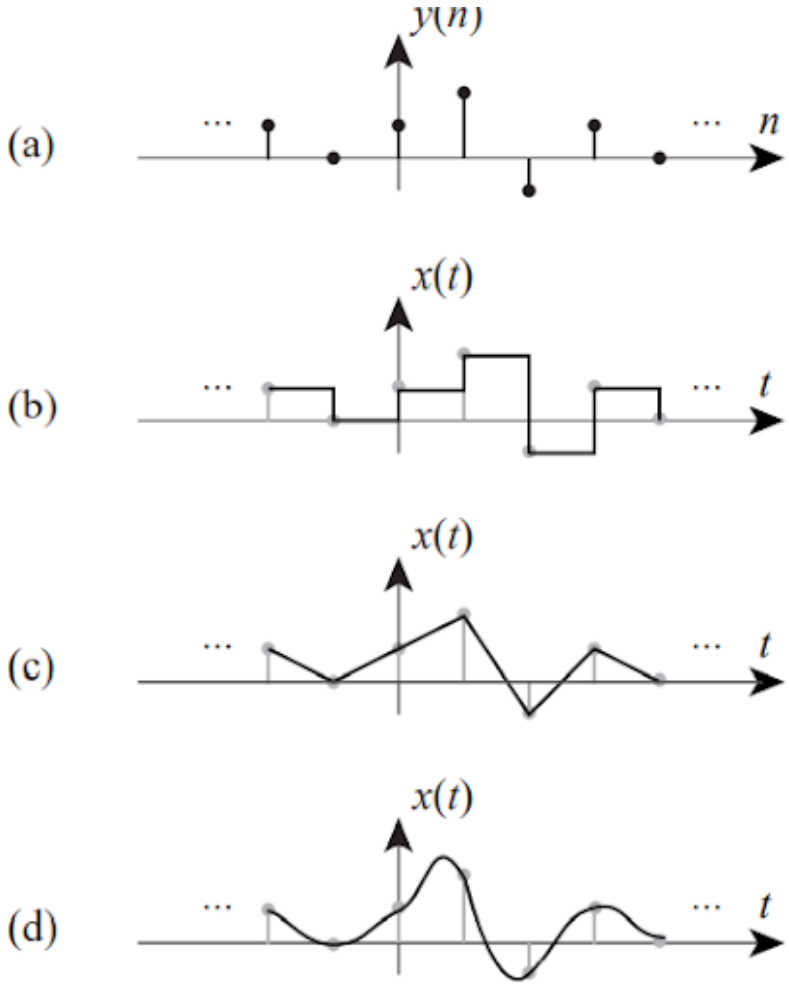


Figura 4: ejemplo práctico de posibles reconstrucciones (b), (c) y (d) de una onda de sonido a partir de una señal discreta en puntos sampleados en (a).
Fuente: [Lee y Varaiya, 2003].

2.1.3. TRANSFORMADA DE FOURIER Y DFT

La transformada de Fourier es una operación matemática que interpreta a una función como si fuese una suma de ondas periódicas de distintas frecuencias y fases, cada una con una amplitud distinta [Bracewell y Bracewell, 1986].

La ecuación 1 es una interpretación de la transformada de Fourier, que expresa a la función $f(x)$ como una suma de infinitas funciones periódicas. La periodicidad está representada por la expresión que eleva a e.

(1) $$f(x) = \int_{-\infty}^{\infty} F(k) e^{2\pi i k x} dk$$

Es necesario mencionar que a pesar de que la transformada de Fourier se define como la suma de infinitas ondas para expresar una función, es posible generar una aproximación de la misma al sumar un número discreto de ondas suficientemente grande. En otras palabras, lo del "infinito" no es necesario. Esto último se conoce como una **transformada de Fourier discreta (DFT)**[Chaudhary, 2020].

(2) $$x(n) = \sum_{k=0}^{K-1} X'(k) e^{j\frac{2\pi}{N}kn} \quad n = 0, 1, 2, ...$$

2.1.4. STFT Y FRAME SIZE

Si se quiere obtener una representación más significativa de un sonido hay que tener en consideración la duración de este. Entre más dure el sonido, más difícil será encontrar una representación de transformada de Fourier que se aproxime a lo que se busca (pues la onda será más compleja), por lo que se suele dividir a los sonidos en ventanas de pequeña duración en tiempo o *"frames"* y aplicar DFT a cada *frame* por separado. Esto se conoce como transformada de Fourier a corto plazo o **STFT** (*short time Fourier transformation*)[Veen, 2013].

Se define al *frame size* como el valor que representa la duración de un *frame*. Considerando que esta operación suele ser aplicada en computadoras que guardan sonido mediante sampleo, el *frame size* se suele representar como la cantidad de *samples* contenidos en cada frame.

2.1.5. HOP SIZE

Hop size, o tamaño de salto, es el intervalo que hay entre los principios de dos *frames* consecutivos [Charif *et al.*, 2010]. De manera similar al *frame size*, se representa en la cantidad de samples contenidos en el intervalo.

2.1.6. ESPECTROGRAMAS

Un **espectrograma** es una representación visual de una STFT aplicada a un extracto de audio. La dimensión horizontal de un espectrograma representa al tiempo y la dimensión vertical

corresponde a frecuencias dentro de un rango. Al aplicar DFT a cada *frame* de lo que dura el audio (dimensión horizontal), se obtienen sumas de ondas periódicas dentro de un rango de frecuencias (dimensión vertical). El espectrograma representará la amplitud de estas ondas periódicas, por lo que si un segmento “brilla” en un espectrograma quiere decir que en el periodo de tiempo indicado por la posición horizontal del segmento, el sonido se compone mayoritariamente por las frecuencias que se muestran en la posición vertical del segmento. La figura 5 corresponde al sonido de un violín representado en un espectrograma.

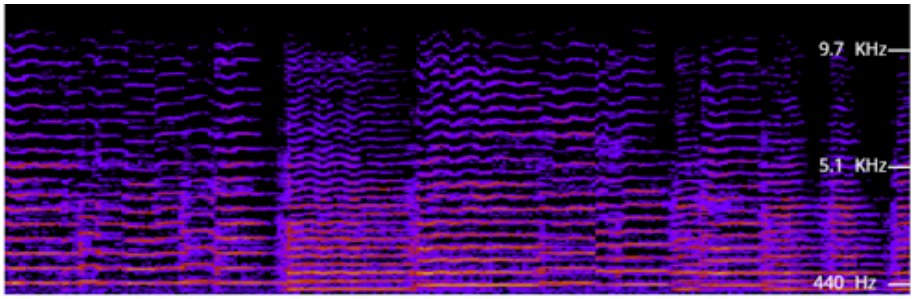


Figura 5: Espectrograma de un violín. Al observar el espectrograma, se observa que la mayor parte de las frecuencias que componen al sonido del violín en toda su duración a través del tiempo se encuentran entre los 440 Hz y los 5.1 kHz.
Fuente: [Academo, 2016].

2.1.7. ESCALA Y ESPECTROGRAMA DE MEL

La tonalidad de un sonido depende de su frecuencia, pero los seres humanos no distinguen estas frecuencias en una escala lineal [Roberts, 2020]. Por ejemplo, la cuarta nota “Do” de un piano (C4) suena a una frecuencia aproximada de 261.63[Hz], la quinta “Do” de un piano (C5) suena a una frecuencia de 523.25[Hz] y la sexta “Do” (C6) suena a 1046.50[HZ] [Suits(1998)]. Como se puede observar, entre dos notas “Do” la frecuencia se multiplica por 2, por lo que la percepción de las notas musicales con respecto a la frecuencia de los sonidos no es lineal.

La escala de **mel** busca representar el sonido de una forma más lineal, básicamente aplicando una operación logarítmica a las frecuencias. Un ejemplo de aplicación de la escala de mel se ilustra en la figura 6.

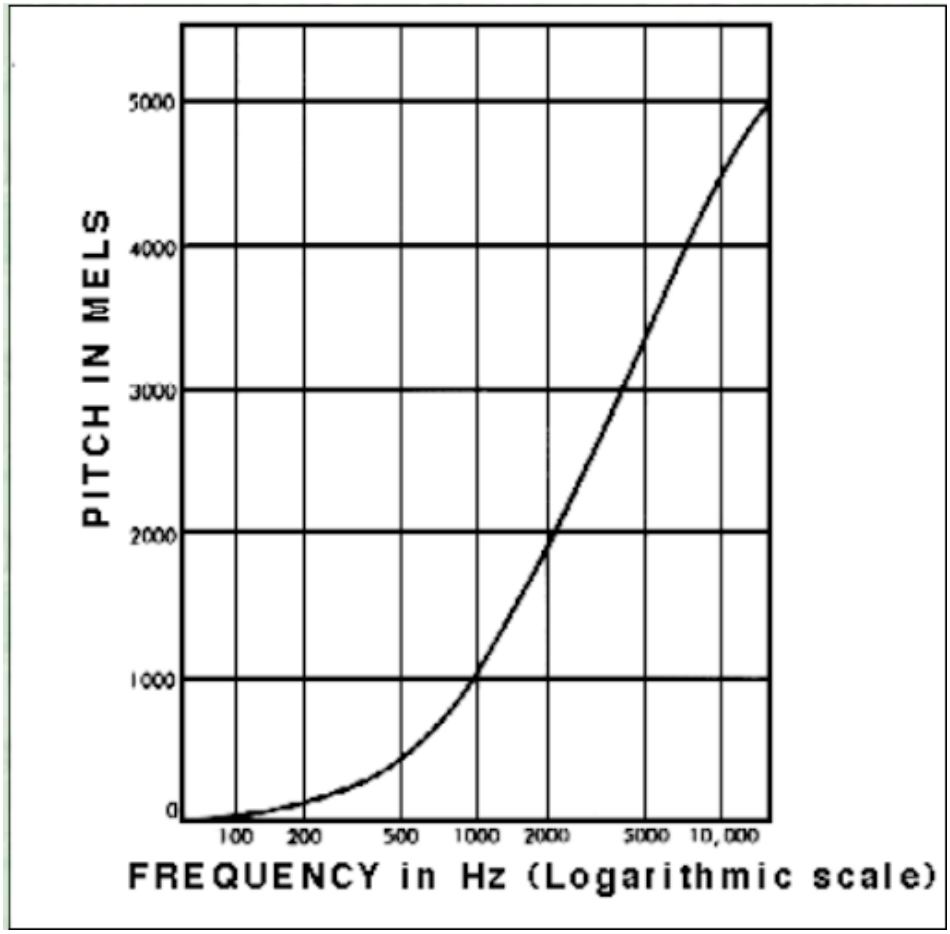


Figura 6: Escala de mel como una función de la frecuencia
Fuente: [Appleton y Perer, 1975].

Un **espectrograma de mel** es un espectrograma en el que el rango de frecuencias se expresa de manera no lineal, sinó que de acuerdo a la escala de mel. Un ejemplo de espectrograma de mel es la figura 7.

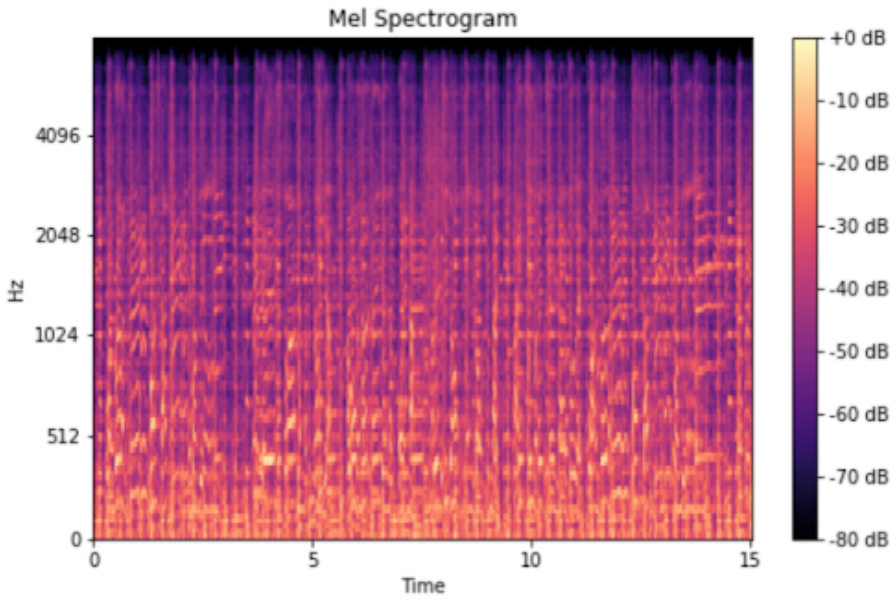


Figura 7: Ejemplo de un espectrograma de mel. Nótese que entre los 512[Hz] y los 1024[Hz], y entre los 1024[Hz] y los 2048[Hz] las distancias visualmente son las mismas, a pesar de que las diferencias en [Hz] no lo son.
Fuente: [Roberts, 2020].

En los espectrogramas de mel la dimensión vertical pasa de frecuencias a *"mels"*. El **número de mels** corresponde a la cantidad de mels en el rango vertical de un espectrograma. Entre mayor sea el número de mels, más precisa será la representación del espectrograma pero más grande (y computacionalmente pesado) será el mismo.

2.2. MODELOS DE PREDICCIÓN

2.2.1. DEFINICIÓN BÁSICA

Los **modelos de predicción** son procesos matemáticos utilizados para predecir eventos o resultados futuros al analizar patrones de un conjunto de datos históricos de entrada. Los modelos de predicción han sido utilizados para predecir ventas, distinguir correos electrónicos de tipo *spam*, estimar la probabilidad de que alguien haga click en un anuncio dentro de una página web, etc [Lawton, 2022].

Como cualquier proceso matemático, los modelos de predicción se suelen implementar mediante algoritmos computacionales.

2.2.2. REDES NEURONALES ARTIFICIALES

Las **redes neuronales artificiales**, o **ANN's** (*artificial neural networks*) son modelos de predicción inspirados en algunos de los procesos detrás del funcionamiento del cerebro humano, concretamente del traspaso de información a través de las neuronas. En este caso, una red neuronal artificial consiste de "nodos" (que actúan como las neuronas) que tienen como misión procesar y traspasar a otros nodos información de entrada X para transformarla en una salida y' , que debe aproximarse lo más posible a un valor esperado y, que puede ser un vector o un escalar.

Durante una "fase de entrenamiento", los nodos ajustan sus parámetros de procesamiento de información según un error calculado entre las salidas (y') y los valores esperados (y) que se va pasando a través de los nodos, a través de un algoritmo llamado *"back propagation"* [Grossi y Buscema, 2008]. La idea es que luego del *back propagation* el error se reduzca en el futuro.

Comúnmente los nodos de una ANN se organizan en "capas ocultas". La información de entrada X se procesa en los nodos de la primera capa, luego el resultado se pasa a la segunda capa, se procesa, se pasa a la tercera capa y así sucesivamente hasta llegar a una capa de salida desde la cual se obtiene y'. Durante el *back propagation*, el error se pasa al revés: se "reparte" el error entre los nodos de la última capa para que ajusten sus parámetros de procesamiento, luego se reparte el error a los nodos de la penúltima capa, etc. Un ejemplo simple de esta arquitectura se ilustra en la figura 8.

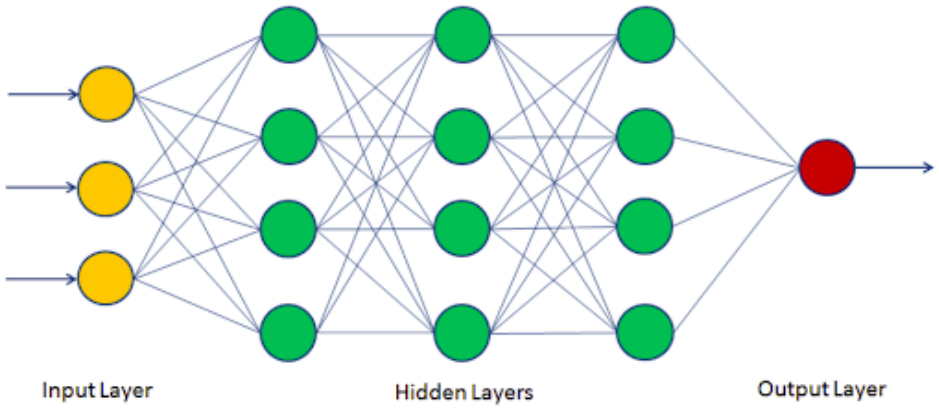


Figura 8: arquitectura simplificada de una ANN. Consiste en la capa de entrada a la izquierda, las capas ocultas al centro y la salida a la derecha.
Fuente: [Navlani, 2019].

2.2.3. CONJUNTOS DE ENTRENAMIENTO, VALIDACIÓN Y PRUEBAS

Para preparar y evaluar un modelo como una ANN se necesitan datos de ejemplo, los cuales se organizan en tres conjuntos: el **conjunto de entrenamiento**, el **conjunto de validación** y el **conjunto de pruebas** [Alpaydin, 2020].

El **conjunto de entrenamiento** corresponde a los datos con los que se entrena el modelo. Todo el proceso de **back propagation** para ajustar los parámetros de los nodos se hace usando elementos de este conjunto.

El **conjunto de validación** son los ejemplos con los que se evalúa el desempeño del modelo *durante* el entrenamiento. Típicamente el proceso de entrenamiento se puede dividir en distintas iteraciones o *epochs*. Luego de cada *epoch*, se evalúa el modelo "actual" usando ejemplos que no han participado del entrenamiento. Estos ejemplos conforman al conjunto de validación. Luego de esta evaluación se continúa el proceso de entrenamiento en el siguiente *epoch* con los datos del conjunto de entrenamiento y así sucesivamente.

El **conjunto de pruebas** son los ejemplos con los que se evalúa el modelo una vez finalizado el proceso de entrenamiento. Los resultados de esta evaluación deberían servir como una métrica del desempeño del modelo en un caso real.

2.2.4. REDES NEURONALES CONVOLUCIONALES Y MAPAS DE CARACTERÍSTICAS

Las **redes neuronales convolucionales** o CNN's son un tipo de ANN's que han sido utilizadas principalmente para clasificación de imágenes [Yamashita *et al.*, 2018]. Las CNN's cuentan con neuronas especiales capaces de procesar "convoluciones", es decir, patrones dentro de espacios típicamente bidimensionales o unidimensionales.

Por ejemplo, para el procesamiento de imágenes (caso bidimensional), los atributos de entrada de una imagen corresponden a los píxeles que la conforman. Sería útil buscar patrones conformados por grupos de píxeles cercanos (como formas geométricas) en lugar de considerar a cada píxel como independiente de los demás.

Para lograr esto, cada convolución cuenta con un *"kernel"*, que en el caso bidimensional es una matriz de tamaño $K \times K$. El kernel va tomando "segmentos" de tamaño $K \times K$ de la entrada y realiza combinaciones lineales entre los elementos de cada segmento y los elementos del kernel, como en el ejemplo ilustrado en la figura 9. En el caso de procesamiento de imágenes, la figura 10 muestra el resultado de aplicar convoluciones en una foto. La figura 10 es un ejemplo de mapas de características, visualizaciones que describen qué partes de la imagen son resaltadas por las convoluciones.

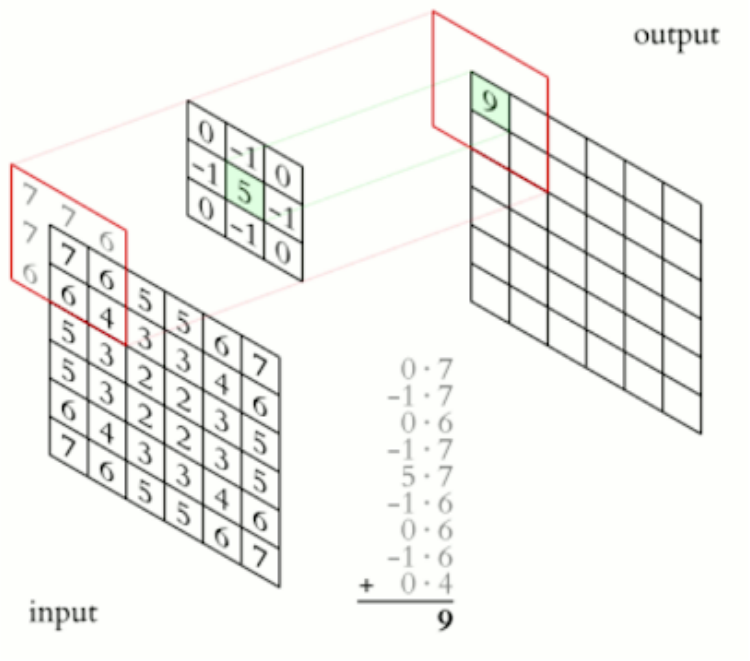


Figura 9: Ejemplo de procesamiento de un kernel en una CNN. En este caso, el kernel es una matriz de tamaño 3×3, cuyos elementos se multiplican por los de un segmento de la entrada de tamaño 3×3 y se suman en la matriz de salida dando un valor de 9. El mismo kernel repetirá el ejercicio para los otros segmentos de tamaño 3×3 de la entrada hasta completar la matriz de salida.
Fuente: [Ganesh, 2019].

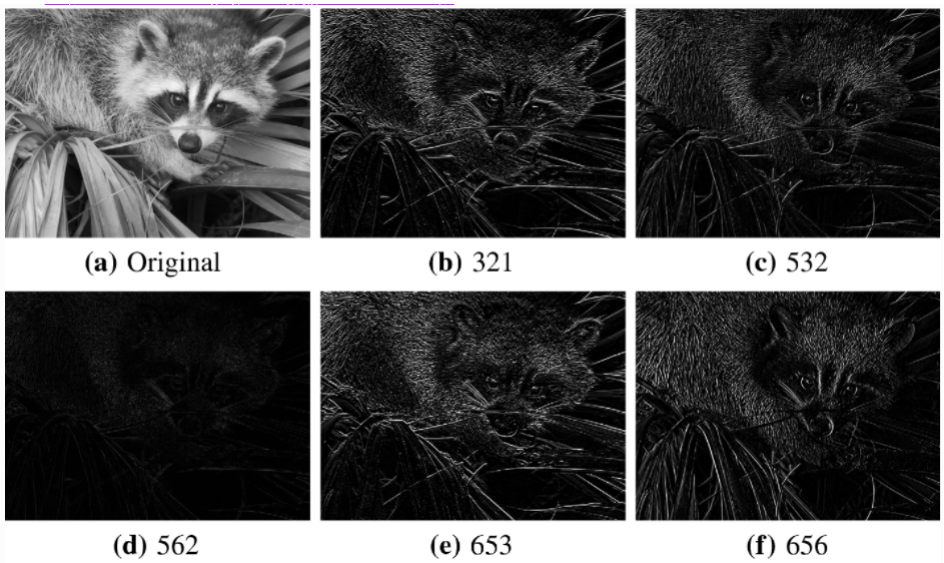


Figura 10: Ejemplo de aplicación de varias convoluciones a la imagen de un mapache. Notar que en los resultados resaltan patrones como los bordes del mapache, su pelaje o los bordes de las plantas en las que se recuesta.
Fuente: [Capobianco *et al.*, 2021].

Durante el entrenamiento de una CNN, los elementos de cada kernel se van modificando. También es necesario aclarar que se pueden procesar varias convoluciones en una misma entrada de manera consecutiva. En este caso, cada convolución se aplicaría en el resultado de la convolución anterior.

2.3. CLASIFICACIÓN MULTIETIQUETA

El **problema de clasificación multi-etiqueta** consiste en identificar al conjunto de etiquetas a las que corresponde cada una de las instancias de entrada dentro de un grupo de instancias. A diferencia de la clasificación **multi-clase**, también llamada clasificación de etiqueta única (*single label*), se considera el caso de que una instancia pueda ser clasificada por múltiples etiquetas en lugar de una sola [Read y Pérez-Cruz, 2015].

Un ejemplo de problema de clasificación multi-etiqueta es determinar qué objetos se pueden identificar en una imagen de entrada, mientras que en un planteamiento acorde al problema multi-clase, se pedirá determinar cuál es el objeto principal visible en la imagen. Otro ejemplo es el presente en la figura 11, que indica cuatro tópicos distintos que pueden estar presentes en varios textos: Deportes, Religión, Ciencias y Política.

Textos (d) \ Etiquetass (l)	Deportes	Religión	Ciencias	Política
Texto 1	x			x
Texto 2			x	x
Texto 3	x			
Texto 4		x	x	

Figura 11: Ejemplo de situación multi-etiqueta.
Fuente: [Alfaro, 2021].

Existen diversas estrategias para resolver problemas de clasificación multi-etiqueta, las cuales se dividen en dos categorías: **"transformación del problema"** y **"adaptación del modelo"** [Tsoumakas y Katakis, 2009].

Los enfoques basados en **transformación del problema** buscan, valga la redundancia, transformar el problema de multi-etiqueta a multi-clase. Uno de estos métodos se llama **"relevancia binaria"** o **"BR"** y consiste en identificar la presencia o ausencia de cada una de las posibles etiquetas de manera separada. De esta forma, si se tienen L posibles etiquetas, existirán L clasificadores independientes. La figura 12 ilustra BR aplicada al ejemplo de la figura 11.

Textos \ Etiqueta	Deportes	No deportes
Texto 1	X	
Texto 2		X
Texto 3	X	
Texto 4		X

Textos \ Etiqueta	Religión	No religión
Texto 1		X
Texto 2		X
Texto 3		X
Texto 4	X	

Textos \ Etiqueta	Política	No política
Texto 1	X	
Texto 2	X	
Texto 3		X
Texto 4		X

Textos \ Etiqueta	Ciencias	No ciencias
Texto 1		X
Texto 2	X	
Texto 3		X
Texto 4	X	

Figura 12: Ejemplo de relevancia binaria.
Fuente: [Alfaro, 2021].

Otro método basado en transformación del problema es **"label powerset"** o **"LP"**, que consiste en considerar a subconjuntos de etiquetas presentes en los datos como si fueran clases separadas. La figura 13 ilustra una aplicación de LP al ejemplo de la figura 11.

Etiqueta / Ejemplo	Deportes	Deportes y Política	Ciencias y Política	Ciencias y Religión
Texto 1		x		
Texto 2			x	
Texto 3	x			
Texto 4				x

Figura 13: Ejemplo de label powerset.
Fuente: [Alfaro, 2021].

Por otro lado, los enfoques basados en adaptación de modelo consisten en adaptar el clasificador para que pueda trabajar directamente con casos multi-etiqueta. Por ejemplo, se puede reentrenar y modificar una red neuronal artificial para que en lugar de clasificar un dato de entrada eligiendo a una de entre L clases, sea capaz de elegir a múltiples clases (que en este caso pasarían a denominarse "etiquetas").

2.3.1. COMMITTE MACHINES

"Committee machines" es una estrategia que consiste en utilizar múltiples modelos de de predicción independientes o "miembros de un comité" y combinar las predicciones que éstos entreguen en una sola estimación [Tresp, 2001]. Hay muchas formas de combinar las predicciones del comité, como por ejemplo promediar los resultados. También se puede usar cada miembro del comité para predecir un elemento distinto del resultado, en caso de que este sea multi-dimensional.

2.4. MÉTRICAS DE EVALUACIÓN

2.4.1. CLASIFICACIÓN BINARIA

Se considera que un modelo es de clasificación binaria cuando este debe clasificar a cada dato de entrada en una de dos clases. Generalmente una de las clases es considerada como "positiva" y la otra como "negativa". Por ejemplo, en un modelo que diagnostica una enfermedad en un paciente, la clase positiva correspondería al caso en el que el paciente padece de la enfermedad y la clase negativa, el caso contrario.

Ante este escenario, se obtienen cuatro valores que se obtienen luego de predecir un conjunto de datos:

- TP (*"true positives"*): el número de ejemplos **positivos** del conjunto que fueron **clasificados correctamente como positivos**

- FP *("false positives")*: el número de ejemplos **negativos** del conjunto que fueron **clasificados incorrectamente como positivos**
- TN *("true negatives")*: el número de ejemplos **negativos** del conjunto que fueron **clasificados correctamente como negativos**
- FN *("false negatives")*: el número de ejemplos **positivos** del conjunto que fueron **clasificados incorrectamente como negativos**

2.4.2. ACCURACY

"*Accuracy*" es de las métricas más utilizadas en la evaluación de modelos de clasificación. Consiste en la razón de ejemplos correctamente clasificados como positivos o negativos en el total de ejemplos, de acuerdo con la ecuación 3.

(3) $$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

2.4.3. PRECISSION

"*Precission*" es una métrica correspondiente a la razón entre el número de predicciones positivas correctas y el número total de predicciones positivas, según la ecuación 4.

(4) $$Precission = \frac{TP}{TP + FP}$$

Si *precission* es bajo, el modelo está identificando incorrectamente ejemplos negativos como si fuesen positivos de manera recurrente.

Esta métrica no es perfecta, pues si se clasifica a un solo dato del conjunto de pruebas como positivo y ese dato termina siendo positivo, la *precission* será de 1 o un 100 %, aunque hayan muchos más ejemplos positivos en el conjunto de pruebas que fueron clasificados como negativos. Por lo tanto, *precission* no es una métrica que pueda evaluar correctamente el desempeño de un modelo de clasificación por si sola [Log, 2021].

2.4.4. RECALL

"*Recall*" es una métrica correspondiente a la razón entre el numero de predicciones positivas correctas y el número total de ejemplos positivos del conjunto, según la ecuación 5.

(5) $$Recall = \frac{TP}{TP + FN}$$

Si el *recall* es bajo, el modelo no está identificando correctamente ejemplos positivos de manera frecuente.

Al igual que *precission*, *recall* no es una métrica perfecta, pues si se clasifican a todos los elementos del conjunto de pruebas como positivos, el *recall* será de 1 o un 100 %, aunque todos los ejemplos negativos del conjunto de pruebas hayan sido clasificados incorrectamente como positivos. Por lo tanto, *recall* tampoco es una métrica que pueda evaluar correctamente el desempeño de un modelo de clasificación por si sola.

2.4.5. F1-SCORE

Del *trade-off* que existe entre *precission* y *recall* [Bennett, 2020] nace la motivación de encontrar una métrica que tome lo mejor de las dos. Esta métrica es *"f1-score"*, que corresponde a la media armónica entre *precission* y *recall*, según la ecuación 6. La media armónica es útil pues "penaliza" muy bien los casos en los que precission o recall sean muy bajos.

(6) $$\textit{f1-score} = 2 * \frac{\textit{Precission} * \textit{Recall}}{\textit{Precission} + \textit{Recall}}$$

F1-score resulta muy util en casos en los que exista mucho imbalance de clases [Bhatt, 2018].

2.4.6. MEDIDAS MACRO Y MICRO

F1-score, *precission* y *recall* son medidas aplicables a casos de clasificación binaria. De acá surge una duda: ¿cómo se pueden adaptar estas medidas a casos multi-clase o multi-etiqueta?.

Esta problemática puede abordarse desde dos enfoques distintos, el enfoque "macro" y el enfoque "micro" [Kumar, 2020].

El enfoque "macro" aplica una medición por cada clase o etiqueta y entrega el promedio aritmético de las mediciones. Por ejemplo, para obtener el *f1-score macro* en un caso multi-clase o multi-etiqueta, se saca el "f1-score" de cada clase o etiqueta "c" de un conjunto C y se promedia. Las ecuaciones 7, 8 y 9 permiten obtener el *f1-score*, *precission* y *recall* desde el enfoque macro.

(7) $$\textit{f1-score macro} = \frac{\sum_{c \in C} \textit{f1-score}_c}{|C|}$$

(8) $$\textit{precission macro} = \frac{\sum_{c \in C} \textit{precission}_c}{|C|}$$

(9) $$\textit{recall macro} = \frac{\sum_{c \in C} \textit{recall}_c}{|C|}$$

Una desventaja del enfoque macro es que "asume" de manera implícita que todas las clases o etiquetas tienen el mismo nivel de relevancia en el resultado final. Por lo tanto, si existe un desbalance en la cantidad de muestras de cada clase o etiqueta dentro del conjunto C, el enfoque macro no será representativo. Para estos casos se plantea el enfoque "micro", en el cual se toman las muestras de todas las clases en conjunto para calcular directamente el resultado, en lugar de calcular separadamente por clases o etiquetas. Para que se entienda mejor, las ecuaciones 10 y 11 muestran como calcular *precission micro* y *recall micro*, donde TP_c, FP_c, TN_c y FN_c son los cuatro valores explicados en la sección 2.4.1, calculados para cada clase o etiqueta $c \in C$.

(10) $$\textit{Precission micro} = \frac{\sum_{c \in C} TP}{\sum_{c \in C} TP + \sum_{c \in C} FP}$$

(11) $$\textit{Recall micro} = \frac{\sum_{c \in C} TP}{\sum_{c \in C} TP + \sum_{c \in C} FN}$$

F1-score micro se calcula de forma análoga a como se calcula *f1-score* en el caso binario, de acuerdo con la ecuación 12.

(12) $$\textit{f1-score micro} = 2 * \frac{\textit{Precission micro} * \textit{Recall micro}}{\textit{Precission micro} + \textit{Recall micro}}$$

2.5. MATRIZ DE CONFUSIÓN

Una **matriz de confusión** es una herramienta utilizada para medir el desempeño de un modelo de predicción en problemas binarios o multi-clases. Juan Ignacio Barros [Arce, 2019] la describe así: "*Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. En términos prácticos, la matriz de confusión permite ver qué tipos de aciertos y errores está teniendo el modelo a la hora de pasar por el proceso de aprendizaje con los datos*".

En el caso de un modelo de predicción binario, los valores de la matriz serán TP, FP, TN y FN. La figura 14 representa este caso.

Figura 14: Matriz de confusión binaria
Fuente: [Arce, 2019].

La figura 15 ilustra una matriz de confusión en un caso multi-clase de tres clases: A, B y C. A modo de ejemplo, E_{BA} corresponde a la cantidad de veces en las que el modelo predice incorrectamente a un ejemplo de la clase A como parte de la clase B.

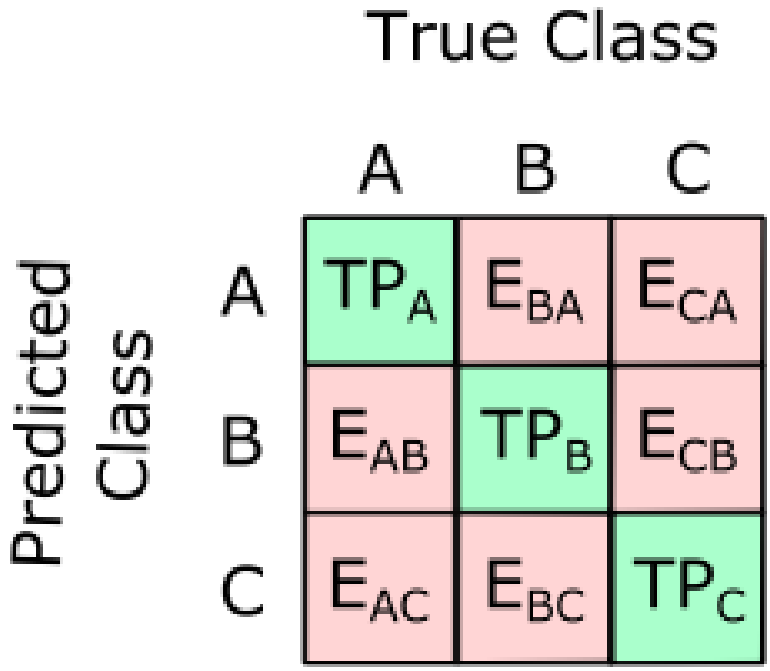


Figura 15: Matriz de confusión multi-clase, en el caso de 3 clases
Fuente: [Tharwat, 2018].

El caso óptimo en cuanto a las predicciones del modelo ocurre cuando la matriz de confusión es una matriz diagonal, pues el modelo no interpretó erróneamente a ningún ejemplo de ninguna clase, entregando valores no nulos sólo en la diagonal de la matriz.

CAPÍTULO 3
PROPUESTA DE SOLUCIÓN

En esta sección se describe el proceso realizado para desarrollar los modelos de predicción. Desde la preparación de los conjuntos de datos que se utilizarán y el pre-procesamiento de estos mismos hasta la preparación de los modelos en sí.

3.1. PLATAFORMA

Para la realización de este trabajo se hace uso de Google Colab [Google, 2022], una plataforma de Google que permite programar en servidores remotos distintas tareas como análisis de datos y desarrollo de modelos de aprendizaje automático gracias a que estos servidores cuentan con hardware especializado. En este caso se utiliza el plan "pro" de Google Colab, que ofrece más recursos que el plan gratuito.

Prácticamente todo el desarrollo de la parte práctica de este trabajo se realiza usando el lenguaje python 3.6. Algunas de las librerías de python más importantes que se utilizan son:

- *tensorflow 2.X*[Abadi *et al.*, 2015], para el desarrollo de los modelos de aprendizaje
- *numpy 1.21*[Harris *et al.*, 2020] para el trabajo con vectores de datos de gran tamaño
- *librosa 0.9.1* para el procesamiento de archivos de audio[McFee *et al.*, 2015]

3.2. OBTENCIÓN DE BASE DE DATOS MULTIETIQUETA

Multi-clase a multi-etiqueta

La base de datos de canciones utilizada para este trabajo se obtiene a partir de otra base de datos ya existente "O" extraída desde Spotify [Spotify, 2018] con un número de 77563 canciones [Apolo, 2022]. La lista O está manualmente clasificada por géneros musicales en modalidad multi-clase, es decir, un género musical por canción. Por lo tanto, la primera tarea es extraer un conjunto de datos multi-etiqueta a partir del conjunto multi-clase original.

Como se debe tomar en cuenta tanto el peso total de los datos que los servidores de *Google Colab* pueden procesar como el tiempo que tomará trabajar con todos estos datos, se debe trabajar con un subconjunto de O en lugar de O en su totalidad. Con el fin de que el subconjunto final esté balanceado en cuanto a sus géneros, a partir de O se selecciona de manera

aleatoria un conjunto de canciones P_c por cada clase c perteneciente a O, donde $|P_c|$ se obtiene de la ecuación 13, siendo O_c el subconjunto de canciones de O que pertenecen a la clase c. Todos los conjuntos P_c conforman al "conjunto de preselección" o P.

(13) $$|P_c| = \text{mín}(350, |O_c|)$$

Para cada P_c se asigna el 60 % de sus canciones a lo que será el conjunto de entrenamiento, un 20 % al de validación y el 20 % restante al de pruebas conformando respectivamente a los conjuntos P_{train}, P_{val} y P_{test} que juntos forman a P. Posteriormente se borran las canciones duplicadas en y entre los 3 conjuntos resultantes, es decir, las canciones en las que se repitan el nombre del artista y el nombre de la canción (pues hay veces en las que un artista sube a su biblioteca una misma canción múltiples veces). Luego de este proceso, los conjuntos P_{train}, P_{val} y P_{test} terminan con un total de 22137, 6990 y 7082 canciones, respectivamente.

Para pasar de conjunto multi-clase a uno multi-etiqueta con respecto a los géneros, se hace uso de la API que Spotify provee [Spotify, 2018], la cual permite obtener los múltiples géneros musicales que corresponden a una canción. Es necesario aclarar que los géneros musicales de una canción de Spotify están determinados exclusivamente por los artistas y que todas las canciones de un artista compartirán los mismos géneros musicales. Por lo tanto, desde ya se debe trabajar con el supuesto de que un artista solo produce música perteneciente a un conjunto preestablecido de géneros musicales y que este conjunto está presente de manera invariante en todas sus canciones (por supuesto esto no es así y podría afectar el desempeño final de los modelos entrenados).

Selección de géneros musicales y eliminación de duplicados

El conjunto multi-etiqueta P incluye en sus canciones un total de 2887, 2296 y 2318 géneros distintos para P_{train}, P_{val} y P_{test}, respectivamente. Es difícil entrenar un modelo que identifique cantidades tan grandes de géneros, por lo que es más práctico seleccionar un conjunto de $n \geq 30$ géneros para clasificar.

Así se elige al conjunto G, conformado por 33 géneros musicales seleccionados, presentes tanto en P_{train}, en P_{val} y en P_{test}. Estos géneros se muestran en la tabla 1, junto a la cantidad de apariciones de estos en P_{train}, P_{val} y P_{test}.

Tabla 1: Géneros musicales seleccionados del conjunto P.

Fuente: Elaboración Propia.

Género	Apariciones en P_{train}	Apariciones en P_{val}	Apariciones en P_{test}
trap latino	517	118	131
pop rock	455	106	131
reggeaton	442	112	127
classical	433	127	132
argentine rock	355	115	104
house	354	135	121
rock-and-roll	352	82	104
death metal	352	111	108
soul	329	78	98
electropop	307	93	102
bossa nova	292	85	104
psychedelic rock	289	97	103
chilean rock	281	85	75
blues rock	275	79	99
filmi	274	89	88
samba	261	78	82
hip hop	257	89	102
funk	248	75	69
r&b	246	59	67
quiet storm	235	56	72
children's music	235	83	85
drum and bass	223	79	71
post-teen pop	218	62	46
dubstep	218	66	80
vocal jazz	217	70	67
breakbeat	217	69	69
gothic metal	212	53	58
nu jazz	200	58	55
k-pop	198	64	61
synthpop	194	68	60
punk	194	62	76
tango	183	57	60
disco	172	52	57

Estos géneros fueron seleccionados según los siguientes criterios:

- Se evitó seleccionar géneros muy ambiguos o que puedan englobar muchos tipos de música, como *"rock"*, *"pop"* o *"latin"*. Muchos de estos géneros también tuvieron de-

masiadas apariciones en P, factor que dificultaría, en caso de incluirlos, el balanceo de datos en el futuro.

- Se intentó tomar en cuenta la variabilidad en los géneros seleccionados, incluyendo subgéneros de rock, metal, pop, jazz, música electrónica, etc.
- Se seleccionaron géneros específicos a países como *tango* (Argentina), *samba* (Brazil), *filmi* (India) o *k-pop* (Corea), asumiendo que estos tienen las características suficientes para ser reconocidos por los clasificadores que se desarrollarán en el futuro.
- Considerando el punto anterior, se incluyó a los géneros *chilean rock* (rock chileno) y *argentine rock* (rock argentino) por razones experimentales. Se busca comprobar si es posible para un clasificador distinguir estos dos géneros usando solo el audio de sus canciones.
- Hay que aclarar que se seleccionó al género *trap latino* en lugar de *trap* pues la cantidad de apariciones de *trap* en P fue muy escasa en comparación a su variante latina (solo 41 apariciones en P_{train}).

Una vez hecha la selección de géneros que conforman a G, se descartan de P todas las canciones que no incluyan a ninguno de los géneros pertenecientes a G. De esta forma, el conjunto P se filtra resultando en el conjunto D, conformado por el conjunto de entrenamiento D_{train} (P_{train} filtrado) con 7598 canciones, el de validación D_{val} (P_{val} filtrado) con 2268 y el de pruebas D_{test} (P_{test} filtrado) con 2386.

Para entender las relaciones entre los géneros seleccionados, la figura 16 presenta una matriz de coocurrencias de los géneros en todo el conjunto D. Esta matriz indica el ratio de apariciones de cada género dentro del conjunto de canciones en las que un género específico aparece. Por ejemplo, en la primera fila de la matriz se puede ver que en el 81 % de canciones en las que el género “trap latino” aparece, el género “reggeaton” también está presente. Se puede notar que, si bien esta matriz no es simétrica, tiende a la simetría. Siguiendo con el ejemplo anterior, el trap latino aparece en un 84 % de canciones en las que el reggeaton está presente. Si bien este número no es igual al del caso inverso, el nivel de apariciones es similar en ambos casos. Este patrón se suele repetir en muchos otros pares de géneros, entre los cuales están:

- *trap latino* y *reggeaton* (0.81-0.84)
- *soul* y *quiet storm* (0.45-0.59)
- *funk* y *soul* (0.68-0.56)
- *quiet storm* y *disco* (0.66 - 0.52)
- *disco* y *funk* (0.51-0.37)

- *bossa nova* y *samba* (0.69-0.6)

Estos pares de géneros hacen sentido según el contexto cultural en los que se hicieron populares. El *reggeaton* y el *trap latino* son géneros que han ganado mucha popularidad en latinoamérica dentro de lo que se considera como “pop urbano”, influenciando el primero al segundo [Díaz, 2018]. El *soul*, el *funk*, la música *disco* y el *quiet storm* fueron algunos de los géneros que se hicieron relevantes dentro de la cultura afroamericana entre la década de los 50’s y los 70’s ([musicforwardfoundation.com, 2021], [Harvey, 2012]) . Por último, tanto el *bossa nova* como la *samba* son géneros de la música popular brasileña que surgieron el el siglo XX [bnbmusiclessons.com, 2015].

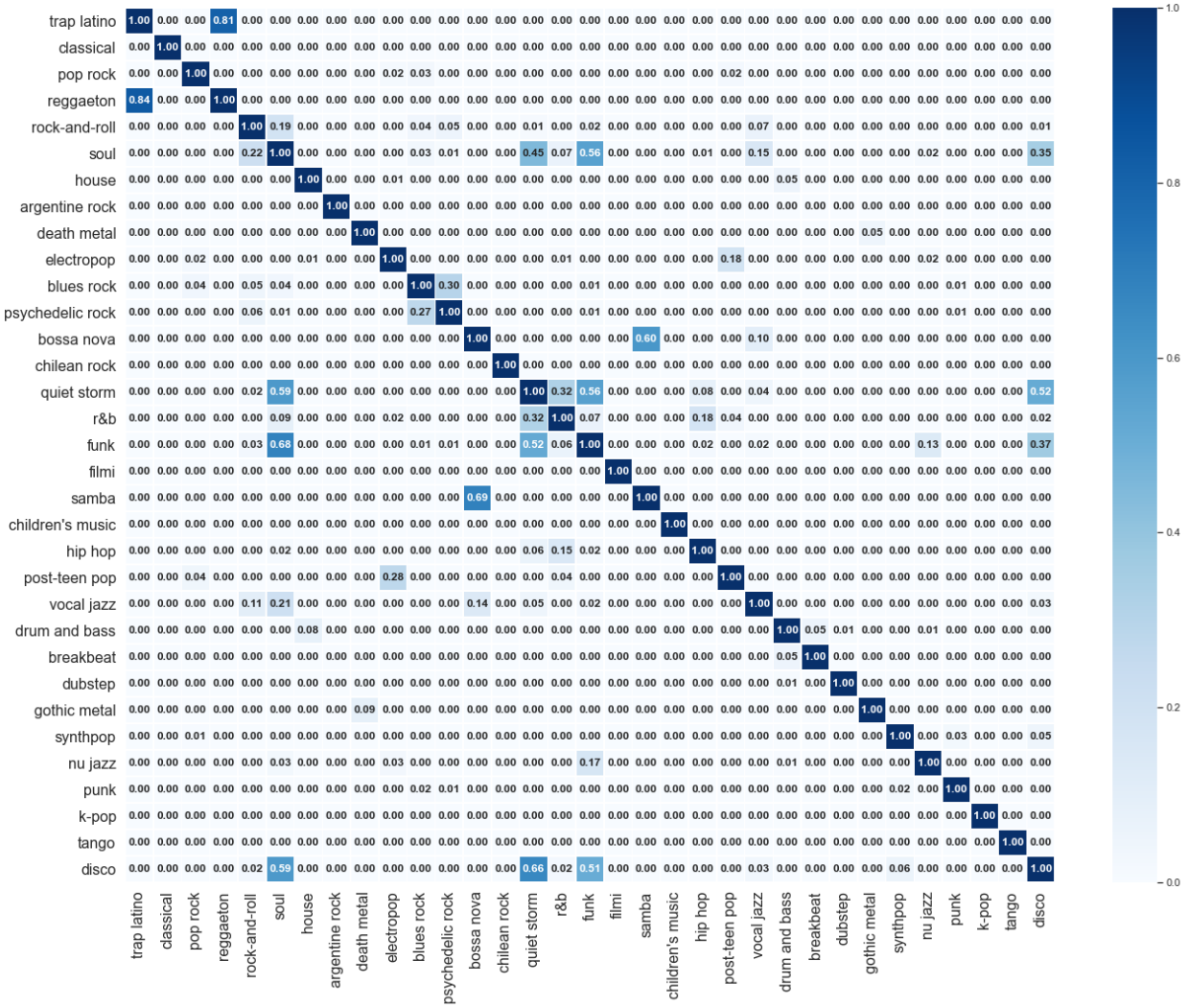


Figura 16: Matriz de coocurrencias de géneros en D

Por otro lado, la figura 16 indica que también hay muchos otros géneros musicales con pocas o nulas relaciones simultaneas con los demás géneros en cuanto a apariciones conjuntas. Llama la atención por ejemplo que *chilean rock* y *argentine rock* no compartan apariciones con otros subgéneros del rock como el *pop rock* o incluso el *punk* a pesar de que la influencia de estos géneros pueda estar presente en canciones de rock chileno o argentino. A su vez, *children's music* (música para niños) no comparte ninguna relación con los otros géneros de la lista, a pesar de que sus canciones pueden perfectamente tomar elementos de estos (por ejemplo, una canción de una película de Disney puede sonar a *pop-rock* y seguir siendo considerada como música infantil solo por ser de Disney), pues lo que distingue a este género no es tanto sus características instrumentales sino el rango etario de su público objetivo. Todos estos factores podrían dificultar la tarea de identificar los géneros de una canción usando un modelo predictor.

Balanceo de datos

El desbalance de clases es un problema que suele afectar a modelos de clasificación [Ling y Sheng, 2010]. Este problema ocurre cuando algunas clases tienen mayores frecuencias de aparición que otras en el conjunto de datos desde donde se entrena el clasificador. Tratar este problema en el caso multi-clase no suele ser difícil pues en muchos casos se pueden generar nuevas muestras sintéticas a partir de los datos pertenecientes a las clases menos frecuentes usando, por ejemplo, el algoritmo *"SMOTE"* [Brownlee, 2020]. Sin embargo, métodos como *SMOTE* no son tan útiles en el caso multi-etiqueta [Daniels y Metaxas, 2017]. Para este trabajo, se opta por utilizar los algoritmos *ML-ROS* y *ML-RUS*[Charte *et al.*, 2015].

Al igual que *SMOTE*, *ML-ROS* es un algoritmo de *oversampling*, es decir, consiste en duplicar de manera artificial datos. Lo que distingue a *ML-ROS* es la elección de los ejemplos que se duplicarán, tomando en cuenta el caso multi-etiqueta. El método para muestrear datos que se representan en audio es explicado en la sección 3.5. El proceso que se describe a continuación, en cambio, corresponde a la decisión de los datos que se duplicarán en nuevas muestras y el número total de muestras que se crearán para cada dato.

En este caso, para cada etiqueta g del conjunto G presente en un conjunto de datos D_{actual} se calcula el $IRLBl_g$ o "ratio de imbalance de la etiqueta", valor que se calcula dividiendo la cantidad de apariciones de la etiqueta más frecuente del conjunto de datos por la cantidad de apariciones de g según la ecuación 14, donde $d_i \in D_{actual}$ corresponde al i-ésimo ejemplo de D_{actual}. Este valor será igual a 1 para la etiqueta más frecuente y mayor a 1 para el resto de etiquetas.

$$IRLBl(g) = \frac{argmax_{g' \in G} \sum_{i=1}^{D} h(g', d_i)}{\sum_{i=1}^{D} h(g, d_i)}, h(g, d_i) = \begin{cases} 1 & \text{si } y \in d_i \\ 0 & \text{si } y \notin d_i \end{cases} \tag{14}$$

Una vez calculado $IRLBl$ para cada $g \in G$, se calcula el promedio de estos valores o $meanIR$. De este modo, todas las etiquetas g con $IRLBl_g > meanIR$ se consideran "minoritarias", es decir, tienen pocas apariciones en el conjunto de datos. Para cada etiqueta minoritaria se obtiene su $minBag$ o "bolsa de mínimo", es decir, el subconjunto de datos en D_{actual} en los que la etiqueta está presente. En cada iteración, se selecciona de manera aleatoria un ejemplo de cada uno de los $minBags$ obtenidos y se duplica. Una vez duplicado un ejemplo por cada $minBag_g$, se recalculan los $IRLBl$'s y $meanIR$ (tomando en cuenta los datos duplicados) y se descartan a los $minBags$ de las etiquetas con $IRLBl_g \leq meanIR$ (pues la etiqueta ya no es minoritaria). Se repite el proceso hasta que se hayan descartado todas las $minBags$ o se hayan duplicado $SamplesToClone(P)$ datos (ver ecuación 15), siendo $P > 0$ un hiperparámetro.

(15) $$SamplesToClone(P) = P * \frac{|D_{actual}|}{100}$$

Para este ejemplo en concreto se proponen dos modificaciones a *ML-ROS*. La primera es que todo el proceso se repite un número de $epochs > 0$ veces. Luego de cada $epoch$ se vuelven a elegir los $minBags$ al recalcular los $IRLBl$'s de todas las etiquetas y $meanIRLBl$. Las segunda modificación es que la selección aleatoria de los ejemplos que se duplicarán por cada $minBag_g$ sigue un patrón probabilístico, donde la probabilidad de que un dato $d_i \in minBag_g$ sea escogido para ser duplicado aumente con el número de etiquetas minoritarias presentes en d_i y disminuya con el número de etiquetas no minoritarias presentes (ver ecuación 18) y la cantidad actual de muestras del dato (es decir, el dato original más sus copias) de acuerdo con la ecuación 16, que depende también de hiperparámetros $\alpha_o \geq 1$ y $\beta_o \geq 1$ según la ecuación 17.

(16) $$Prob_o(d_i, minBag_g) = \frac{Q_o(d_i)}{\sum_{d_j \in minBag_g} Q_o(d_j)}$$

(17) $$Q_o(d_k) = \frac{1}{Penality(d_k)^{\alpha_o} * numeroDeCopias(d_k)^{\beta_o}}$$

(18) $$Penality(d_k) = \frac{1 + |g' \in d_k, IRLBl_{g'} \leq meanIR|}{1 + |g' \in d_k, IRLBl_{g'} > meanIR|}$$

De este modo, el pseudocódigo del proceso es el siguiente:

procedure ML-ROS-MODIFICADO(D_{actual},$epochs$,P,α_o,β_o)
  $SamplesToClone \leftarrow P * \frac{|D_{actual}|}{100}$
  $e \leftarrow 0$
  while $e < epochs$ **do**:
    $i \leftarrow 0$
    for each $g \in G$ **do**:
      if $IRLBl_g > meanIR$ **then**:
        $minBag_{i++} \leftarrow Bag_g$
      end if
    end for
    while $SamplesToClone > 0$ and $i > 0$ **do**
      for each $minBag_g \in minBags$ **do**
        d_k $\leftarrow$ seleccionar dato de $minBag_g$ seleccionado con probabilidad $Prob_o(d_k, minBag_g)$
        $d'_k \leftarrow$ Duplicar d_k
        agregar d'_k a D_{actual}
        $SamplesToClone - -$
      end for
      for each $minBag_g \in minBags$ **do**
        if $IRLBl_g \leq meanIR$ **then**:
          remover $minBag_g$ de $minBags$
          $i - -$
        end if
      end for
    end while
    $e + +$
  end while
end procedure

Luego de aplicar *ML-ROS-MODIFICADO* a los conjuntos de datos que se tienen se aplica adicionalmente el algoritmo *ML-RUS*. A diferencia de *ML-ROS*, *ML-RUS* es un algoritmo de *undersampling*. Es decir, en lugar de duplicar datos cuyas etiquetas aparezcan poco en el conjunto, se eliminan datos cuyas etiquetas aparecen mucho.

El proceso de *ML-RUS*, junto con las modificaciones que se proponen usar, es similar a *ML-ROS-MODIFICADO*. El cambio principal es que ahora se elije descartar datos pertenecientes a $maxBags$, conjuntos de datos en los que se encuentran las etiquetas mayoritarias $g \in G$, donde $IRLBl_g \leq meanIR$. Se considera que las etiquetas mayoritarias tienen demasiadas apariciones en el conjunto de datos D_{actual}.

Al igual que en *ML-ROS-MODIFICADO*, se modificó también la probabilidad de que un dato de $maxBag_g$ sea escogido, esta vez para ser eliminado. Aquella probabilidad se obtiene con el inverso de lo que se presenta en la ecuación 17 esta vez usando hiperparámetros $\alpha_u > 0$

y $\beta_u > 0$, resultando en las ecuaciones 19 y 20. Notar también que ahora se consideran a los $maxBags$ en lugar de los $minBags$.

(19) $$Prob_u(d_i, maxBag_g) = \frac{Q_u(d_i)}{\sum_{d_j \in maxBag_g} Q_u(d_j)}$$

(20) $$Q_u(d_k) = Penality(d_k)^{\alpha_u} * numeroDeCopias(d_k)^{\beta_u}$$

Así, el pseudocódigo de este proceso es el siguiente:

---

procedure ML-RUS-MODIFICADO(D_{actual},$epochs$,P,α_u,β_u)
  $SamplesToEliminate \leftarrow P * \frac{|D_{actual}|}{100}$
  $e \leftarrow 0$
  while $e < epochs$ **do**:
    $i \leftarrow 0$
    for each $g \in G$ **do**:
      if $IRLBl_g \leq meanIR$ **then**:
        $maxBag_{i++} \leftarrow Bag_g$
      end if
    end for
    while $SamplesToEliminate > 0$ and $i > 0$ **do**
      for each $maxBag_g \in maxBags$ **do**
        d_k $\leftarrow$ seleccionar dato de $maxBag_g$ seleccionado con probabilidad $Prob_u(d_k, maxBag_g)$
        eliminar d_k de D_{actual}
        $SamplesToEliminate--$
      end for
      for each $maxBag_g \in maxBags$ **do**
        if $IRLBl_g$ $meanIR$ **then**:
          remover $maxBag_g$ de $maxBags$
          $i--$
        end if
      end for
    end while
    $e++$
  end while
end procedure

En esta ocasión, se permite el caso en que algunos datos se eliminen completamente del conjunto de datos (es decir, que queden con 0 muestras en total) luego de aplicar *ML-RUS-MODIFICADO*.

Para este caso en específico se aplica *ML-ROS-MODIFICADO* y *ML-RUS-MODIFICADO* a los conjuntos D_{train} y D_{val}. El conjunto D_{test} no se somete a *oversampling* o *undersampling* pues en el futuro se debe evaluar la capacidad de los modelos predictores que se piensan desarrollar de generalizar al momento de evaluar patrones presentes en el conjunto de pruebas, que debería reflejar distribuciones presentes en un "caso real" sin ningún tipo de manipulación [Santos *et al.*, 2018].

ML-ROS-MODIFICADO se aplica tanto a D_{train} y D_{val} con $P = 8$, $epochs = 5$, $\alpha_o = 2{,}5$ y $beta_o = 5$. Consecutivamente se aplica *ML-RUS-MODIFICADO* con $P = 5$, $epochs = 1$, $\alpha_u = 3$ y $beta_u = 3$ (estos valores han sido ajustados mediante múltiples pruebas hasta alcanzar un balance aceptable).

Las distribuciones de las etiquetas presentes en los conjuntos D_{train} y D_{val}, tanto antes como después de aplicar *ML-ROS-MODIFICADO* y *ML-RUS-MODIFICADO* se muestra en la figura 17. También se muestra las frecuencias de las etiquetas en el conjunto D_{test}, que no fue alterado durante este proceso.

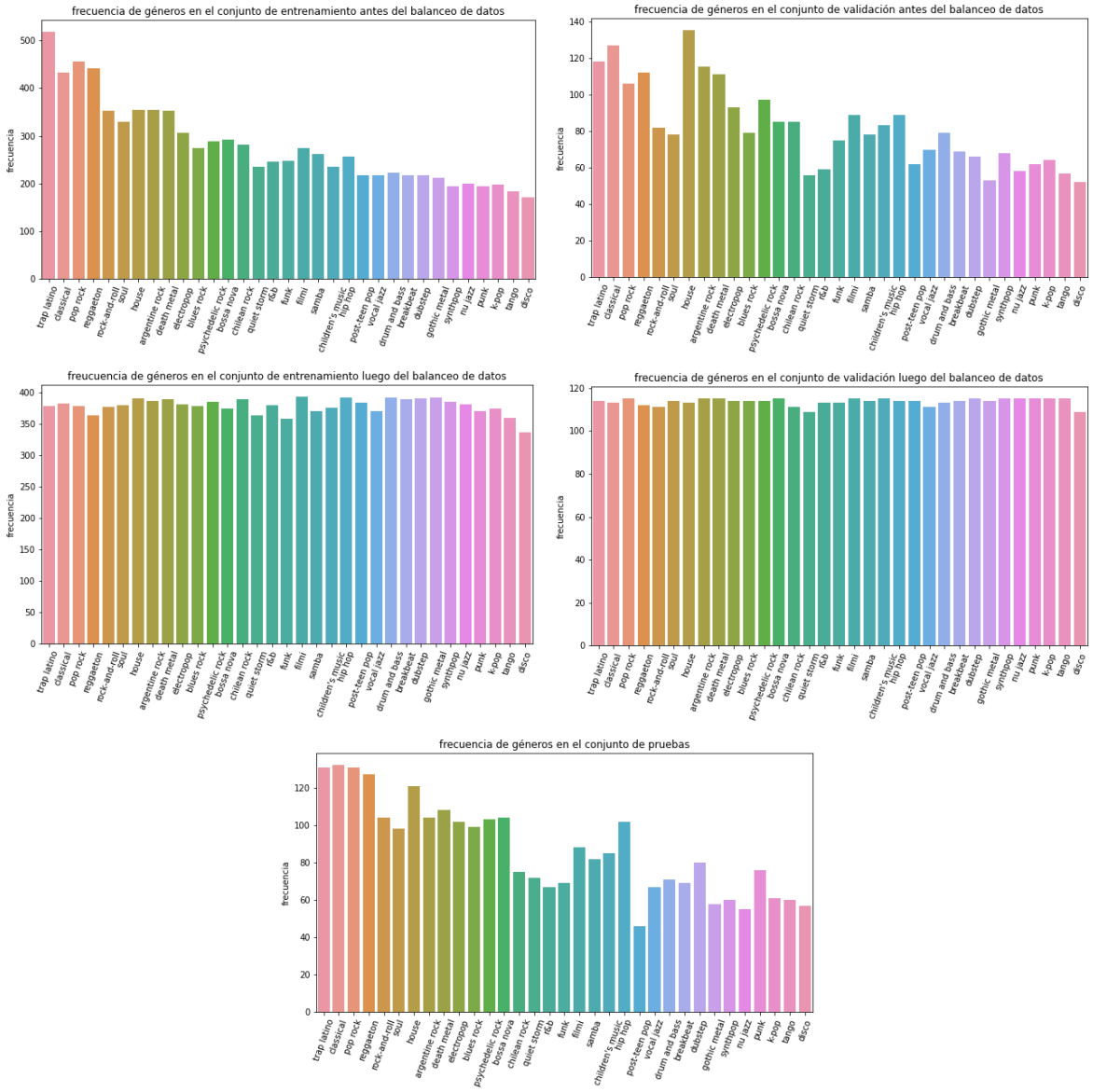


Figura 17: Frecuencias de apariciones de géneros en los conjuntos de canciones antes y después del balanceo de datos. Las frecuencias D_{test} también se muestran. Notar que el balance de apariciones en D_{train} y D_{val} está mucho más equilibrado luego del balanceo de datos.

Si consideramos a cada muestra de una misma canción como una sola muestra, los conjuntos D_{train}, D_{val} y D_{test} quedan con 7299, 2226 y 2386, muestras respectivamente. Notar que para D_{train} y D_{val} el número de muestras disminuyó pues *ML-RUS* se aplicó con la libertad de eliminar canciones completamente. Para D_{test} el número se mantiene igual pues no se aplicó el balanceo de datos.

Si consideramos a cada muestra de cada canción como muestras separadas, los conjuntos

D_{train}, D_{val} y D_{test} quedan con 10287, 3069 y 2386 muestras.

3.3. OBTENCIÓN DE PISTAS DE AUDIO

Cada canción de D (considerando a una sola muestra por canción, ignorando al balanceo de datos) se representa inicialmente por una pista de audio de 30 [s]. Esta pista se obtiene nuevamente haciendo uso de la API de spotify, la cual puede entregar de manera gratuita un fragmento de 30[s] de una canción solicitada [Spotify, 2018].

Cabe mencionar que la elección de los 30[s] de cada canción es hecha por la API y no se puede cambiar, por lo que se debe asumir que esos 30 segundos serán representativos de la canción y sus géneros.

3.4. SEPARACIÓN DE INSTRUMENTOS

Una vez se tienen los conjuntos de canciones D_{train}, D_{val} y D_{test} se propone utilizar una herramienta de separación de instrumentos llamada *demucs* [Défossez *et al.*, 2019].

Demucs es un modelo de procesamiento de audio pre-entrenado capaz de separar (con cierta exactitud) una pista musical en cuatro fuentes de audio: *vocals* (vocalización), *drums* (percusión), *bass* (bajo) y *other* (otros o instrumentalización).

De este modo, se propone usar a *demucs* para separar las pistas de audio de las canciones de D_{train}, D_{val} y D_{test}.

Se puede ver en la figura 18 un paralelismo entre la separación de canciones en fuentes musicales y la separación de imágenes a color en canales RGB (rojo, verde y azul) que se suele utilizar para el preprocesamiento de datos en modelos de predicción en base a imágenes [Castro *et al.*, 2019].

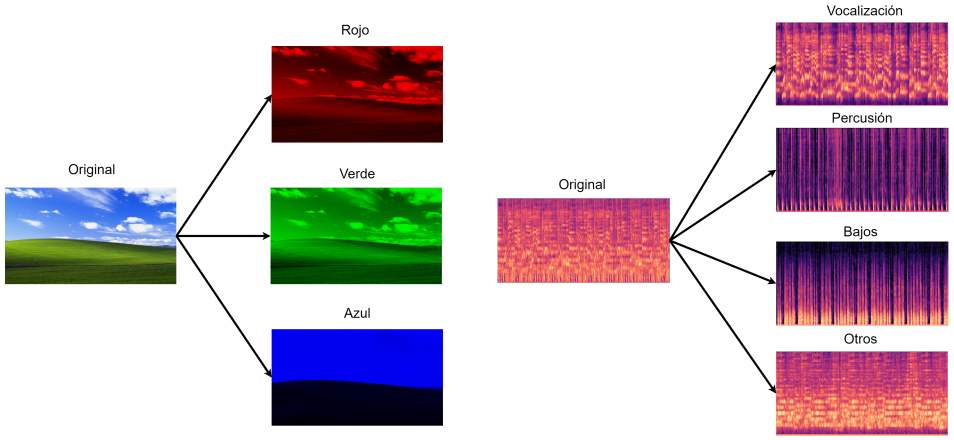


Figura 18: A la izquierda, la separación del salvapantallas de windows xp en canales RGB. A la derecha, la separación de un fragmento visualizado en un espectrograma de mel de 15 segundos de la canción *Never Gonna Give you Up* de Rick Astley en cuatro fuentes musicales.
Fuente: Elaboración propia.

Este es un paso importante dentro de la propuesta que se realiza para esta investigación, pues se espera que un modelo de predicción de géneros musicales logrará un mayor desempeño si se entrena usando canciones separadas por fuentes musicales, en lugar de canciones enteras. Después de todo, un género musical se puede distinguir de manera más efectiva al analizar separadamente el estilo de vocalización que se presenta (como el *soul* o el *hip-hop*), por la caracterización de su ritmo y percusión (como el *reggeaton* o la *samba*), por sus bajos (como el *trap latino* o el *drum and bass*) o por los instrumentos presentes en sus canciones (como la música clásica o el *death metal*).

3.5. MUESTREO Y REPRESENTACIÓN DE ESPECTROGRAMAS

Para que los espectrogramas obtenidos de las múltiples muestras de cada canción no sean exactamente iguales entre sí (y así no comprometer la diversidad de los conjuntos de datos) se aplicó el siguiente método de muestreo: considerando que las pistas de audio de las fuentes musicales de cada canción tienen una duración de 30[s], cada muestra será un extracto aleatorio de 15[s] de sus fuentes musicales.

El pseudocódigo del muestreo es el siguiente:

procedure DUPLICACIÓN($cancion$)
  $vocals \leftarrow$ audio separado de vocalización de $cancion$
  $drums \leftarrow$ audio separado de percusión de $cancion$
  $bass \leftarrow$ audio separado de bajos de $cancion$
  $other \leftarrow$ audio separado de instrumentalización de $cancion$
  $C \leftarrow$ número de muestras de $cancion$
  for each $c \in range(C)$ **do**
    offset $\leftarrow randfloat(0, 15)$
    $vocals_c \leftarrow$ segmento de $vocals$ entre los segundos *offset* y *offset*$+15[s]$
    $drums_c \leftarrow$ segmento de $drums$ entre los segundos *offset* y *offset*$+15[s]$
    $bass_c \leftarrow$ segmento de $bass$ entre los segundos *offset* y *offset*$+15[s]$
    $other_c \leftarrow$ segmento de $other$ entre los segundos *offset* y *offset*$+15[s]$
  end for
end procedure

Ya separadas las fuentes musicales de las canciones, se debe elegir la representación de los archivos de audio resultantes para ser ingresados en los modelos de aprendizaje.

En esta ocasión se opta por utilizar espectrogramas de mel con los siguientes parámetros:

- *sample rate* de audio original = $12[kHz]$
- *frame size* = $512[sample]$
- *hop lenght* = $256[sample]$
- *numero de mels* = $128[mel]$

Esta elección de parámetros se realizó tomando en cuenta las limitaciones de google colab de acuerdo con el tamaño máximo de datos que sus servidores pueden procesar en su plan pro.

Los espectrogramas se generan usando la librería de python *librosa* [McFee *et al.*, 2015] en su versión 0.9.1.

El diagrama 19 ilustra el preprocesamiento de una canción en N muestras, desde la separación de instrumentos hasta la obtención de los espectrogramas. El resultado final son cuatro espectrogramas de dimensiones $128x704$ por muestra.

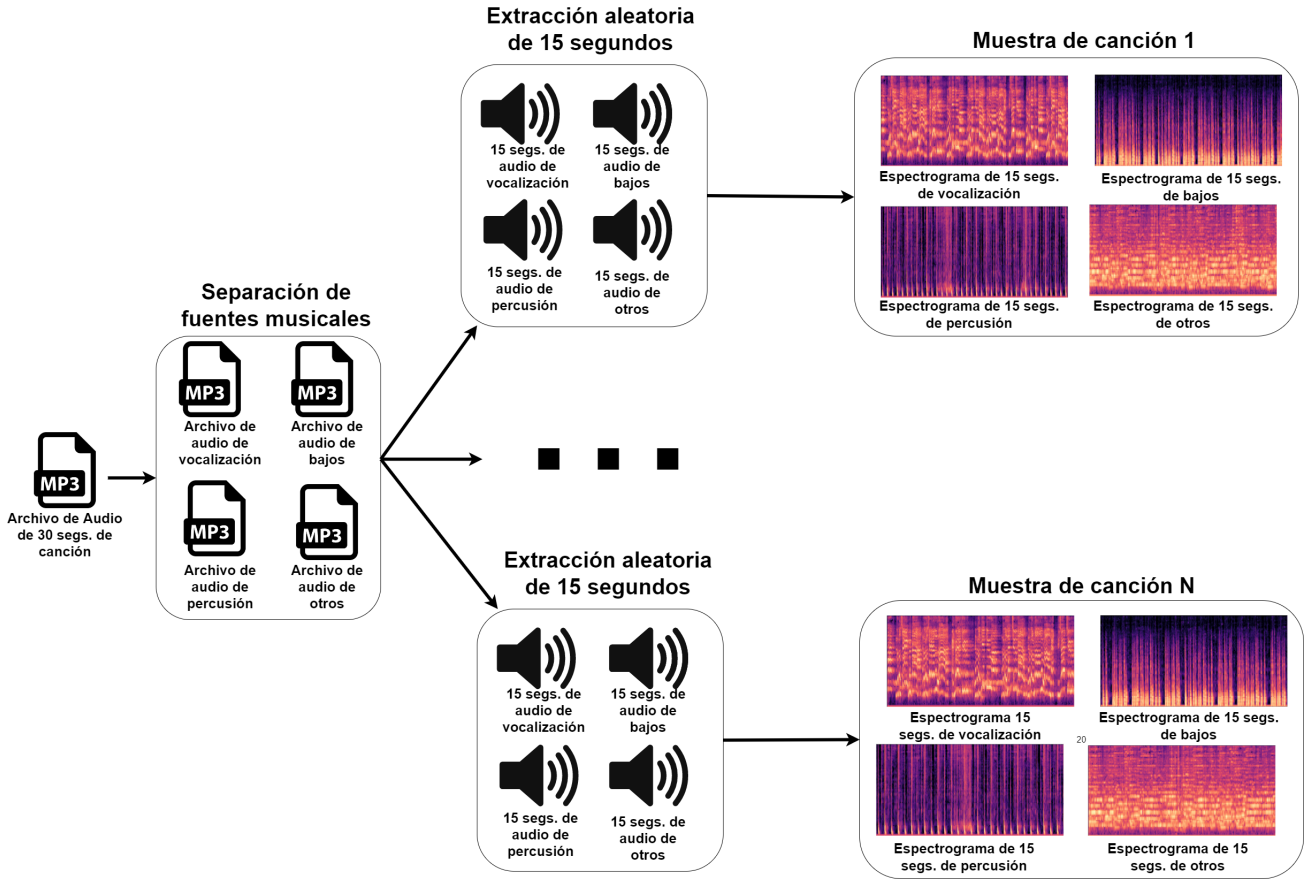


Figura 19: Diagrama del preprocesamiento de una canción individual en N muestras
Fuente: Elaboración propia.

3.6. USO DE REDES NEURONALES CONVOLUCIONALES

Se proponen a continuación dos modelos de aprendizaje basados en redes neuronales convolucionales o CNN's que tomarán como entrada los datos preprocesados en los pasos anteriores.

Ambos modelos se entrenarán usando los conjuntos de datos de entrenamiento y validación con el fin de predecir los géneros musicales presentes en cada canción del conjunto de pruebas una vez terminado el proceso de entrenamiento.

Ambos modelos retornan para cada canción de entrada un vector de probabilidades independientes de largo $|G|$, donde G es el conjunto de géneros seleccionados. Cada elemento $0 \leq p_g \leq 1$ del vector correspondería a la probabilidad de que el género musical $g \in G$ se encuentre en la canción de entrada.

Ambos modelos se desarrollan usando la librería de python *tensorflow* [Abadi *et al.*, 2015]

en su versión 2.X.

3.6.1. MODELO COMPLEJO O CLÁSICO

Arquitectura

Este modelo consiste en una sola red neuronal convolucional entrenada con el fin de predecir a partir de la muestra de una canción un vector de dimensión $|G|$.

La red consta de 4 "subredes" que se aplicarán para cada una de las 4 fuentes musicales de la canción de entrada por separado. Cada subred a su vez consta de 4 capas convolucionales: en la primera capa se aplican 32 convoluciones, en la segunda y la tercera 64 (cada una); y en la cuarta 32. Luego de cada capa se aplica MaxPooling y un dropout con probabilidad de 0.2. Las dimensiones de los kernels en cada capa son de $4x16$ exceptuando el kernel de la cuarta capa, cuyas dimensiones son de $3x3$. Finalmente el resultado de las 4 subredes se combina en una sola capa densa para predecir el vector de dimensión $|G|$. La figura 20 representa a la arquitectura de este modelo.

Esta arquitectura está inspirada en el modelo de predicción basado en CNN's desarrollado por Sergio Oramas, Oriol Nieto, Francesco Barbieri y Xavier Sierra [Oramas *et al.*, 2017], quienes también quisieron abordar el problema de la clasificación multi-etiqueta de géneros musicales. Algunas de las diferencias principales entre este modelo y el propuesto en esta sección es que acá se reduce el número de convoluciones por cada capa (tomando en cuenta las limitaciones de google colab pro) y que las convoluciones se aplican a cada fuente musical por separado en lugar de recibir a la canción completa como entrada.

La decisión con respecto a los tamaños de los kernels se realizó empíricamente, pues el modelo alcanzó un mejor desempeño durante el entrenamiento cuando se eligieron estos tamaños.

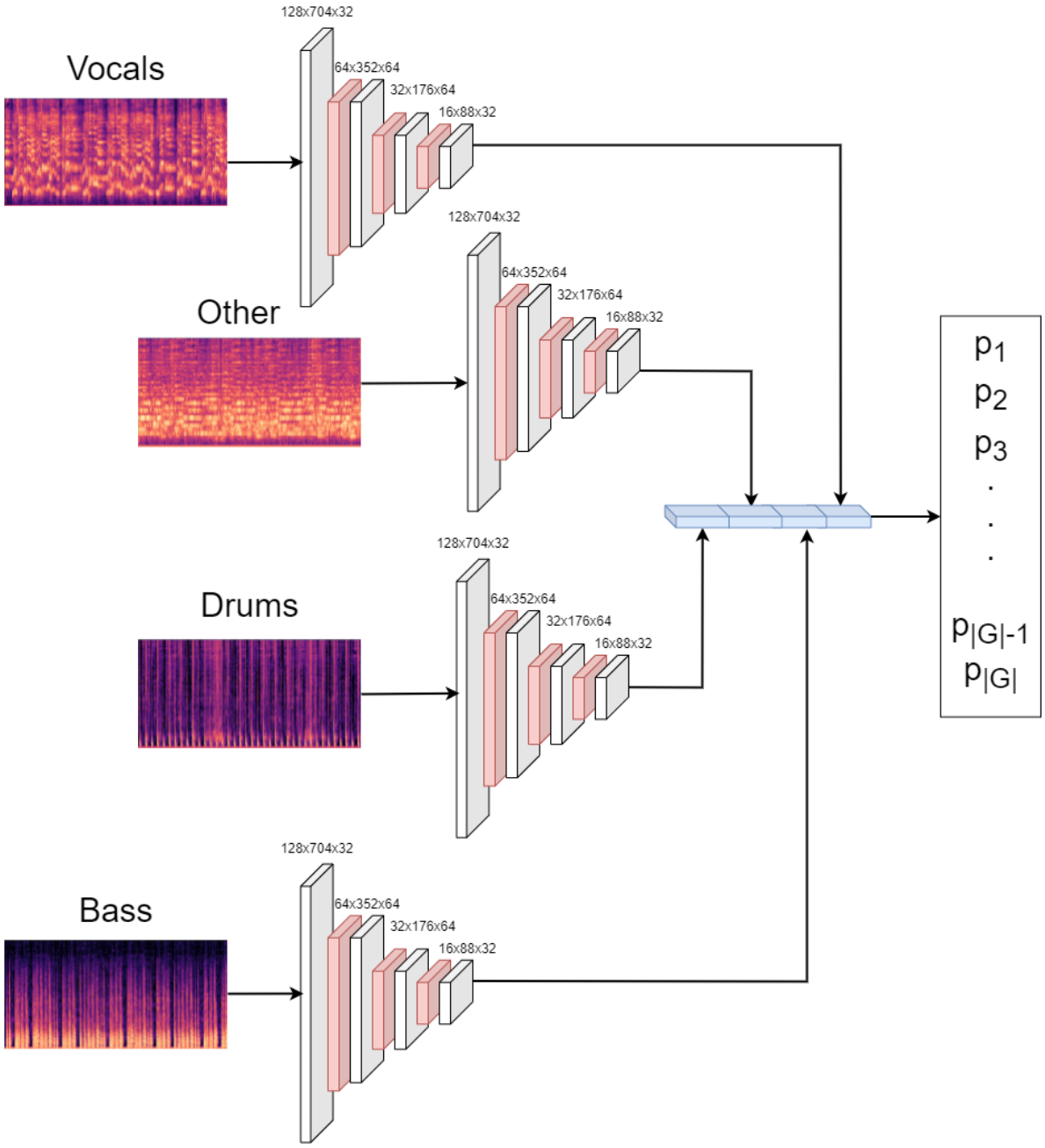


Figura 20: Diagrama de la arquitectura de la red neuronal compleja. Las capas grises representan capas convolucionales, las capas rojizas representan MaxPooling y Dropout y la capa celeste es la capa densa.
Fuente: Elaboración propia.

Entrenamiento

Se plantea usar los conjuntos D_{train} y D_{val} ya obtenidos para entrenar y validar durante el entrenamiento, respectivamente. Esta última afirmación puede resultar obvia, pero en el

modelo alternativo que se explicará luego se utilizarán los conjuntos entrenamiento y validación de una manera distinta.

Formato

Por último, los espectrogramas de D_{train} y D_{val} se procesarán en un formato de flotantes de 16 bits en lugar de los 32 bits que *librosa* usa por defecto. Esta decisión se realiza tomando en cuenta a los límites de memoria de google colab en su plan pro.

3.6.2. COMMITTE MACHINE

Arquitectura

En este caso se propone una solución basada en relevancia binaria para el problema de clasificación multi-etiqueta. En lugar de desarrollar un solo modelo para predecir la presencia de los $|G|$ géneros de cada canción, se entrenan $|G|$ modelos independientes o "miembros de un comité", cada uno con la misión de obtener un p_g para cada $g \in G$. Finalmente, cada p_g es un elemento del vector de probabilidades de salida.

La ventaja de este acercamiento con respecto al de la red neuronal compleja es su escalabilidad, pues en caso de que se quiera agregar un género musical nuevo al modelo basta con crear un nuevo miembro del comité en lugar de reentrenar todo el modelo. Otra ventaja es que cada miembro del comité centra su aprendizaje en detectar un solo género musical, pudiendo alcanzar mejores resultados en una tarea especializada.

Por otro lado, el modelo del comité tiene la desventaja de tener que asumir una completa independencia entre múltiples géneros. Por ejemplo, el reggeaton y el trap latino son géneros altamente relacionados según la matriz de coocurrencias, por lo que al dividir las tareas de identificar ambos géneros por separado se priva al modelo de identificar en las canciones características que los relacionen.

La arquitectura de cada miembro de comité es muy similar a la arquitectura del modelo complejo (4 subredes convolucionales, una para cada fuente) con la diferencia de que ahora se reducen a la mitad el número de convoluciones por capa. Es decir, si antes era 32-64-64-32, ahora es 16-32-32-16. Esta decisión fue tomada con el fin de que cada miembro fuese una versión simplificada de la red del modelo complejo.

La arquitectura del modelo de comité se ilustra en la figura 21. Notar que, como se ha explicado, cada comité es una red neuronal similar al modelo complejo que predice un $p_g, g \in G$.

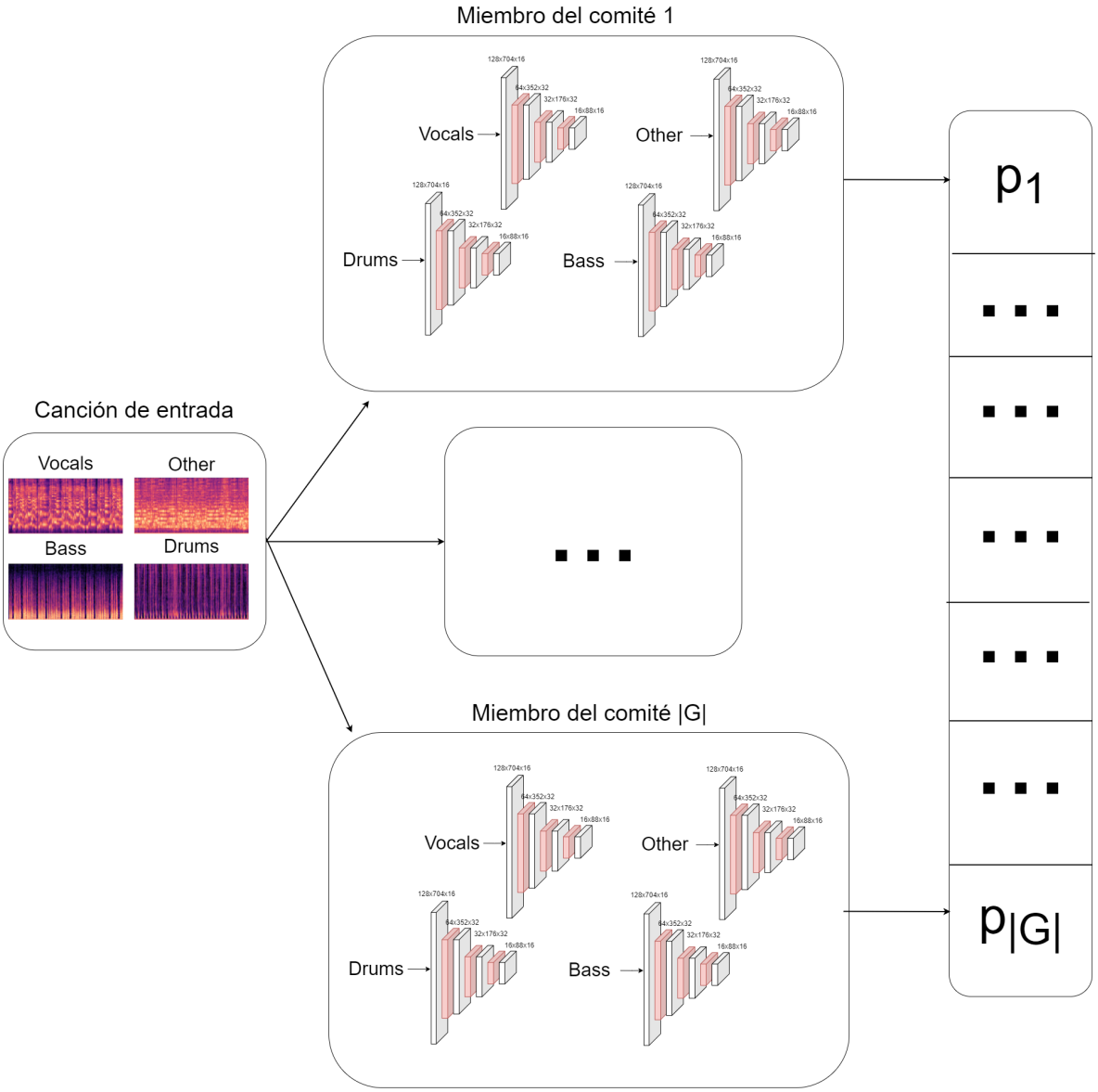


Figura 21: Diagrama de la arquitectura de la red neuronal *committee machine*. Fuente: Elaboración propia.

Entrenamiento

Para el proceso de entrenamiento, cada miembro del comité usará como conjuntos de entrenamiento y validación subconjuntos específicos de D_{train} y D_{val}, respectivamente. Estos conjuntos "D_{train_g}" y "D_{val_g}" son obtenidos según el género musical g ($g \in G$) correspondiente a cada miembro del comité.

El conjunto D_{train_g} se compone de la unión de dos conjuntos de igual tamaño:

1. $\{d : d \in D_{train} | g \in d\}$. En otras palabras, todas las canciones de D_{train} en las que el

género g esté presente.

2. $|\{d : d \in D_{train} | g \in d\}|$ (la cardinalidad del primer conjunto) canciones elegidas aleatoriamente de D_{train} en las que el género g no esté presente.

De este modo, D_{train_g} está compuesto en un 50 % de canciones que tengan a g y un 50 % de canciones que no. Cada miembro del comité hace una predicción probabilística para una tarea de clasificación binaria en específico: identificar si el g correspondiente se encuentra en la canción de entrada o no, por lo que es importante que el conjunto de entrenamiento esté balanceado con respecto a estos dos casos [Ling y Sheng, 2010].

D_{val_g} se obtiene de la misma forma que D_{train_g}, con la diferencia de que se extraen muestras pertenecientes a D_{val} y no de D_{train}. Cada conjunto D_{val_g} estaría compuesto entonces por la unión de estos dos conjuntos:

1. $\{d : d \in D_{val} | g \in d\}$. En otras palabras, todas las canciones de D_{val} en las que el género g esté presente.

2. $|\{d : d \in D_{val} | g \in d\}|$ (la cardinalidad del primer conjunto) canciones elegidas aleatoriamente de D_{val} en las que el género g no esté presente.

Formato

A diferencia del modelo complejo, para el proceso de entrenamiento de cada miembro del comité se usan como entrada espectrogramas en el formato de flotantes de 32 bits por defecto. Al entrenar los miembros por separado, se procesan también los distintos D_{train_g} y D_{val_g} por separado en lugar de los conjuntos completos D_{train} y D_{val}. Al ser estos subconjuntos mucho más pequeños que los conjuntos completos se ahorra mucho en memoria, permitiendo usar un formato de mayor precisión.

CAPÍTULO 4
VALIDACIÓN DE LA SOLUCIÓN

En esta sección se describen y se comentan los resultados en torno a la evaluación de los modelos de clasificación propuestos. Además, se realizará un análisis en torno a algunos de los mapas de características extraíbles de las convoluciones de los modelos.

4.1. evaluación de modelos

Una vez ya entrenados los dos modelos propuestos, el modelo complejo y *committe machine*, estos son evaluados en el conjunto de pruebas D_{test}.

Para comparar los géneros pertenecientes a cada canción con las predicciones hechas por los modelos es necesario considerar los formatos. Los géneros de cada canción t se presentan en un vector binario y_t de largo $|G|$ en el que cada elemento $l_g^t \in y_t$ indica si el género g pertenece a t ($l_g^t = 1$) o no ($l_g^t = 0$). Por otro lado, las predicciones se presentan en un vector de probabilidades independientes y'_t de largo $|G|$. Para cada probabilidad $p_g^t \in y'_t$ se realizará la siguiente transformación:

- Si $p_g^t > 0{,}5$, $p_g^t \to 1$ (se considera que g pertenece a t)
- Si $p_g^t \leq 0{,}5$, $p_g^t \to 0$ (se considera que g no pertenece a t)

4.1.1. métricas de evaluación

Considerando que se está tratando con un problema de detección de muchas etiquetas, se usarán como métricas *f1-score*, *precission* y *recall* tanto en el reconocimiento de cada genero musical por separado como en el de todas las etiquetas en general. Para este último caso se utilizarán los enfoques macro y micro para las métricas de evaluación, considerando que el conjunto de pruebas D_{test} no pasó por el proceso de balanceo de datos.

Estas métricas fueron escogidas pues no solo dan información sobre que tan bien (o mal) son capaces los modelos de predecir correctamente géneros musicales en canciones como lo haría *accuracy*, sino que también pueden dar información sobre los casos en los que se identifican de forma errónea géneros en canciones en las que en realidad no pertenecen (bajo *precission*), o casos en los que no se reconocen géneros en canciones en las que si están presentes (bajo *recall*).

4.1.2. resultados

La tabla 2 presenta los resultados de la evaluación del modelo complejo, tanto en cada uno de los géneros por separado como los resultados generales.

Tabla 2: Resultados de la evaluación del modelo complejo

Fuente: Elaboración Propia.

Género	*f1 score* macro		*precission*		*recall*	
trap latino	0.672		0.539		0.893	
pop rock	0.140		0.203		0.107	
reggeaton	0.675		0.571		0.827	
classical	0.876		0.924		0.833	
argentine rock	0.284		0.347		0.240	
house	0.551		0.540		0.562	
rock-and-roll	0.331		0.491		0.250	
death metal	0.794		0.868		0.731	
soul	0.252		0.400		0.184	
electropop	0.295		0.287		0.304	
bossa nova	0.355		0.402		0.317	
psychedelic rock	0.224		0.400		0.155	
chilean rock	0.075		0.129		0.053	
blues rock	0.142		0.238		0.101	
filmi	0.313		0.390		0.261	
samba	0.364		0.389		0.341	
hip hop	0.530		0.590		0.480	
funk	0.314		0.365		0.275	
r&b	0.325		0.380		0.284	
quiet storm	0.296		0.300		0.292	
childen’s music	0.352		0.439		0.294	
drum and bass	0.529		0.640		0.451	
post-teen pop	0.141		0.132		0.152	
dubstep	0.544		0.551		0.537	
vocal jazz	0.397		0.469		0.343	
breakbeat	0.308		0.248		0.406	
gothic metal	0.574		0.579		0.569	
nu jazz	0.268		0.263		0.273	
k-pop	0.391		0.301		0.557	
synthpop	0.323		0.410		0.267	
punk	0.403		0.444		0.368	
tango	0.644		0.539		0.800	
disco	0.360		0.419		0.316	
TOTAL	macro	micro	macro	micro	macro	micro
	0.395	0.432	0.430	0.461	0.316	0.406

La tabla 3 presenta los resultados de la evaluación del modelo *committee machine*, tanto en cada uno de los géneros por separado como los resultados generales.

Tabla 3: Resultados de la evaluación del modelo commette machines

Fuente: Elaboración Propia.

Género	*f1 score* macro		*precission*		*recall*	
trap latino	0.441		0.291		0.916	
pop rock	0.227		0.139		0.618	
reggeaton	0.465		0.312		0.913	
classical	0.774		0.637		0.985	
argentine rock	0.192		0.109		0.827	
house	0.416		0.273		0.868	
rock-and-roll	0.322		0.216		0.635	
death metal	0.366		0.225		0.991	
soul	0.206		0.128		0.520	
electropop	0.191		0.110		0.755	
bossa nova	0.182		0.101		0.923	
psychedelic rock	0.127		0.091		0.338	
chilean rock	0.135		0.074		0.773	
blues rock	0.190		0.108		0.788	
filmi	0.158		0.087		0.818	
samba	0.147		0.080		0.915	
hip hop	0.212		0.119		0.961	
funk	0.168		0.098		0.609	
r&b	0.204		0.118		0.746	
quiet storm	0.167		0.105		0.403	
childen’s music	0.198		0.113		0.788	
drum and bass	0.232		0.134		0.859	
post-teen pop	0.071		0.037		0.870	
dubstep	0.423		0.275		0.912	
vocal jazz	0.177		0.098		0.866	
breakbeat	0.160		0.087		0.942	
gothic metal	0.179		0.099		0.897	
nu jazz	0.173		0.103		0.545	
k-pop	0.175		0.098		0.869	
synthpop	0.135		0.074		0.800	
punk	0.162		0.088		0.961	
tango	0.304		0.181		0.967	
disco	0.218		0.131		0.667	
TOTAL	macro	micro	macro	micro	macro	micro
	0.240	0.215	0.150	0.124	0.666	0.801

4.1.3. Análisis general

En términos generales ninguno de los modelos alcanzó un desempeño "aceptable". El modelo complejo logró un f-score macro de 0.395 y micro de 0.432 y el modelo del comité apenas alcanzó un f-score de 0.240 y 0.215. Mirando los resultados específicos a cada género se puede ver que si bien algunos géneros se supieron reconocer con mayor exactitud por parte de los modelos como *classical*,*death metal* y *trap latino*, muchos otros géneros dieron bajos resultados en esta evaluación. Se podría decir entonces que la mayoría de géneros seleccionados son poco reconocibles si solo tomamos el audio de sus canciones y que esto afectó significativamente a las medidas generales.

En una disertación realizada por Enric Gauss[Gauss, 2009] se listan algunas de las características que un conjunto de canciones debería tener para facilitar la tarea de crear, a partir del conjunto, un modelo clasificador automático de géneros que sea eficaz. Una de estas características es la variabilidad de cada clase o etiqueta, es decir, que en cada género representado en el conjunto de datos se encuentren canciones diversas. Se desaconseja en ese caso incluir en el conjunto de datos múltiples canciones de un mismo artista o un mismo álbum, ya que se puede presentar un sesgo en el clasificador hacia las características musicales propias del artista o album y no del género. En el caso del conjunto de datos D utilizado para este trabajo (englobando a D_{train}, D_{val} y D_{test}) cada artista aparece en 4 canciones en promedio, con una desviación estándar de 7.87 y 542 de los 2693 artistas (más de una quinta parte) aparecen en más de 5 canciones. Si además de eso se consideran a las canciones duplicadas para el balanceo de datos, se obtiene un conjunto de canciones en el que los artistas se repiten de manera frecuente. Considerando la disertación de Enric Gauss, se podría afirmar este es uno de los factores que influyó de manera negativa en el desempeño de ambos modelos.

Otro factor a tener en cuenta es la elección de los segmentos de audio usados para cada canción. Hay que recordar que se usaron segmentos de 15 segundos seleccionados aleatoriamente de archivos de audio de 30 segundos, los cuales fueron obtenidos usando la API de Spotify con la desventaja de no poder seleccionar el segmento y tener que trabajar por 30 segundos seleccionados directamente por la API. Considerando que este elección fue fija, los segmentos utilizados pudieron no representar correctamente a los géneros de las canciones en muchos casos.

También hay que tener en consideración que los géneros de las canciones se obtuvieron usando la API de Spotify, en la cual los géneros son específicos al artista y no a la canción o al álbum. Por lo tanto, casos en los que un artista decide cambiar su estilo musical durante su carrera pudieron producir confusiones en las predicciones.

Otro factor muy importante a tener en cuenta es que los géneros de cada artista y, por consecuencia, de cada canción de los conjuntos de datos usados para este trabajo fueron elegidos por los mismos artistas o por las personas encargadas de registrarlos en la base de datos de Spotify. Debido a que estas decisiones las realizan seres humanos, son susceptibles a casos

de bajo consenso. Por ejemplo, un artista puede aparecer en la base de datos de Spotify como productor de música *funk*, mientras que otro artista podría no considerarse como un musico de este género aunque produzca música muy similar al primer artista.

No sería justo presentar estos resultados sin realizar alguna comparación con el estado del arte del problema. El trabajo de Sergio Oramas, Oriol Nieto, Francesco Barbieri y Xavier Sierra ([Oramas *et al.*, 2017]) es de los pocos que ha tratado el problema de clasificación de géneros musicales desde la perspectiva multi-etiqueta. En ese trabajo no solo se experimentó con modelos convolucionales basados en audio sino que también se incorporó el análisis de letras de canciones y de portadas de álbumes para el entrenamiento de modelos. Incluso en los modelos convolucionales puramente basados en audio se llegó a muy buenos resultados (alrededor de 0.85, usando la métrica *AUC*).

La pregunta es: ¿porqué ese trabajo fue exitoso en la clasificación de géneros musicales mientras que los resultados de los modelos propuestos para este trabajo no fueron tan fructíferos? La principal respuesta se encuentra en la falta de recursos. En primer lugar, el conjunto de datos usado en [Oramas *et al.*, 2017] es muchísimo mayor en cantidad de canciones (al rededor de 150 mil canciones), además de que los géneros están ligados al álbum y no al artista (permitiendo casos de variabilidad de estilos). En el caso de los modelos presentados en este trabajo, no se pudieron usar conjuntos de entrenamiento muy grandes por las limitaciones de *google colab pro*. Además, para el desarrollo de los modelos propuestos en este trabajo los recursos estuvieron también limitados a lo que *google colab pro* ofrecía. Al experimentar con distintas arquitecturas de redes neuronales convolucionales antes de empezar el entrenamiento oficial de los modelos, el servidor de *google colab pro* frecuentemente dejaba de funcionar por falta de memoria si es que se agregaban más convoluciones o capas a las redes neuronales (especialmente para el modelo complejo, el cual requería de mucho más espacio durante el proceso de entrenamiento), por lo que la complejidad de las arquitecturas escogidas fue limitada. Hay que recordar de la sección 3.6 que la arquitectura de las redes usadas en este trabajo está basada en uno de los modelos usados en [Oramas *et al.*, 2017] pero con la mitad de convoluciones en el caso de la red compleja y un cuarto de las convoluciones en el caso de cada miembro del "comité".

4.1.4. Comparación de los dos modelos

Al comparar los dos acercamientos, el modelo complejo alcanzó resultados mucho mejores que el de comités. Una de las principales razones de porque esto pudo haber pasado se encuentra en los conjuntos usados para preparar ambos modelos.

Hay que recordar que para el modelo complejo se usó D_{train} y D_{val} para el proceso de entrenamiento, conjuntos que en su totalidad tienen una vasta y balanceada cantidad de canciones de cada uno de los 33 géneros seleccionados.

Por otro lado, en el modelo de *committee machine* se usaron subconjuntos de D_{train} y D_{val}

específicos para cada miembro del comité. Cada subconjunto se componía de un 50 % de canciones que pertenecían al genero correspondiente al miembro del comité y un 50 % de canciones que no pertenecían a ese genero. El problema es que ese último 50 % fue formado eligiendo canciones de D_{train} y D_{val} de forma aleatoria, sin ninguna garantía de que incluyera una cantidad representativa de canciones de los otros géneros. También hay que tomar en cuenta que todos los miembros del comité se evaluaron en el conjunto D_{test} en el que cada género musical no es muy numeroso en comparación al total, a diferencia de los subconjuntos de entrenamiento y evaluación, en los que el género correspondiente a cada subconjunto se encontraba en un 50 % de las canciones.

Por lo tanto, cada miembro del comité tuvo las herramientas para aprender a identificar características de canciones que permitían predecir si estas pertenecían a su género asignado, pero no para reconocer los casos en los que ese género no se hacía presente. Esto se evidencia comparando los valores de *precission* obtenidos para cada género en la tabla 3 con los valores de *recall* pues en todos los casos el *recall* es muchísimo mayor. Esto quiere decir que cada miembro del comité predice de forma muy frecuente a las canciones de D_{test} como si fueran parte de su género asignado, incluso en los casos en los que ese género no se hace presente.

Un último comentario con respecto a los subconjuntos de entrenamiento y de validación usados para cada miembro del comité es que cada uno de estos tiene también la desventaja de poseer una cantidad de muestras mucho menor a la de los conjuntos D_{train} y D_{val}, lo cual perjudicó mucho a los resultados finales en el modelo del comité.

4.1.5. Análisis de los resultados en géneros en específico

A continuación, se presentarán breves comentarios sobre los resultados en relación a algunos de los géneros por separado. Para varios de los siguientes comentarios se tomará en cuenta la matriz de coocurrencias representada en la figura 16:

- *classical*: el género cuya evaluación alcanzó los mejores resultados en ambos modelos. Se podría decir que los elementos instrumentales que caracterizan a la música clásica son muy reconocibles en comparación al resto de géneros. La presencia de violines, chelos e instrumentos de viento como trompetas o trombones son algunos de los recursos que raramente se usan en otros géneros musicales más modernos. Por lo tanto, la música clásica es muy identificable en comparación con el resto de géneros.
- *trap latino y reggeaton*: considerando que según la matriz de coocurrencias estos dos géneros aparecen simultáneamente con alta frecuencia, el siguiente comentario aplicará para ambos. Estos dos géneros lograron una puntuación aceptable en los dos modelos, probablemente porque con frecuencia se usan los mismos recursos musicales como el uso de bajos y el reconocible ritmo del *reggeaton*.

- *death metal*: uno de los géneros que alcanzaron mejores resultados en su evaluación en el modelo complejo de acuerdo con la tabla 2. La presencia de guitarras y baterías fuertes y una vocalización caracterizada por los gritos son características que identifican al *death metal*. En el caso del modelo de comités, se alcanzó un *recall* altísimo (0.991) en comparación a *precission* (0.225) según la tabla 3, siendo uno de los muchos casos en los que el miembro del comité no aprendió a identificar canciones que no sean de su género, entregando resultados positivos de manera excesivamente frecuente.

- *gothic metal*: Es necesario tomar en cuenta que el *gothic metal* es un subgénero del *death metal* y que, sin embargo, tiene una relación relativamente baja con este de acuerdo a la matriz de coocurrencias. Por lo tanto, es posible que canciones pertenecientes al *death metal* sean identificadas erróneamente como *gothic metal* y vice versa, aunque considerando que el número de canciones en presentes en estos dos géneros no es muy significativo dentro del total de canciones de D_{test}, es poco probable que este factor haya tenido un impacto relevante en los resultados. En comparación con el *death metal*, hay poco consenso sobre las características que definen al *gothic* metal. Por ejemplo, hay gente que piensa que la inclusión de una voz femenina define a este género y hay gente que no [radio.darkness.com, 2015]. La poca consistencia en cuanto a la definición del metal gótico podría ser una de las razones de porqué los resultados de la evaluación de los modelos en relación a este género sean peores que en relación al *death metal*.

- *pop rock*: Poco desempeño en ambos modelos, y de los pocos casos en los que el modelo *committe machine* alcanzó resultados mejores en comparación con el modelo complejo. Como se puede ver en el nombre de este género, el *pop rock* combina elementos del *pop* y el *rock*. Lamentablemente, tanto el *pop* como el *rock* son géneros extremadamente diversos de los que muchos subgéneros han surgido (muchos de los cuales son parte de los géneros seleccionados para este trabajo). Por lo tanto, es de esperar que exista un bajo consenso sobre los elementos musicales que indiquen cuando una canción pertenece a este género.

- *chilean rock* y *argentine rock*: El rock chileno fue el género cuyo reconocimiento alcanzó el menor desempeño en ambos modelos. Extrañamente, el reconocimiento del rock argentino fue notoriamente mejor, aunque sigue siendo bajo. Estos dos géneros fueron escogidos con fines experimentales, pues se planteaba comprobar si los modelos podían reconocer géneros musicales puramente definidos por el país de origen con solo las características auditivas. Viendo los resultados, la respuesta es negativa.

- *hip hop*: Uno de los géneros cuyo reconocimiento alcanzó la mejor evaluación en el modelo complejo, aunque un 0.530 de *f1-score* (tabla 2) tampoco se podría considerar como “aceptable”. Los ritmos y el estilo de vocalización son característicos del rap. También hay que destacar que este es de los géneros más “únicos” de la lista, pues si no se cuenta al *trap latino* no hay subgéneros del *hip hop* o el *rap* dentro de los 33 géneros escogidos (a diferencia de los múltiples subgéneros de rock, jazz, metal, soul y

música electrónica presentes), factor que pudo favorecer los resultados. En cuanto al modelo *committe machine*, pasa lo mismo que con el *death metal*. Un *recall* de 0.961 y una *precission* de apenas 0.119 indica que el miembro del comité asignado al *hip hop* da muchos resultados positivos incluso en canciones que no son *hip hop*.

- *tango*, *samba* y *bossa nova*: El *tango* es un género musical perteneciente al folclore argentino caracterizado por el uso del acordeón, el piano e instrumentos de cuerda como el violín [Vera, 2011]. Por otra parte, tanto la *samba* como el *bossa nova* son géneros del folclore brasileño caracterizados por estilos tranquilos de guitarra acústica, los cuales si bien son similares tienen sus diferencias. La *samba* y el *bossa nova* están muy relacionados, de hecho, el *bossa nova* es un subgenero de la *samba* [Santiago, 2020]. Se puede corroborar que estos dos géneros brasileños están relacionados según la matriz de coocurrencias, apareciendo simultáneamente en cerca del 65 % de casos (considerar de que la matriz no es simétrica), aunque también hay muchas canciones en las que solo uno de estos dos géneros aparece.

  La razón de porqué se hace mención a estos tres géneros musicales al mismo tiempo es para señalar que, si bien los tres son pertenecientes al folclore de algún país sudamericano, el tango no posee ningún subgénero en la lista de los géneros seleccionados, mientras que la *samba* y el *bossa nova* están relacionados. Considerando este hecho y observando que en el modelo complejo el reconocimiento del *tango* alcanzó un f-score bastante aceptable (0.644) en comparación con el de la *samba* (0.364) y el de el *bossa nova* (0.355) (nótese que ambos valores de f-score son similares), se puede formular la hipótesis de que el modelo complejo tuvo mayor dificultad reconociendo los géneros musicales brasileños que el *tango* debido a que este último es único en la lista y no posee subgéneros, haciendo que sea más fácil de identificar. Por otro lado, hay casos en los que se identifica incorrectamente a una canción de *samba* como de *bossa nova* y vice-versa. En relación al modelo de comités, pasa lo mismo que en el *death metal*, el *hip hop* y muchos otros géneros. Un valor de *recall* muy alto y *precission* muy bajo para los tres géneros.

  Esta hipótesis hace sentido con otros casos. Observar que en ningún subgénero de rock en la lista (incluyendo a *punk*) se alcanza un *f1-score* aceptable en ambos modelos. Pasa lo mismo con los géneros *r&b*, *funk* y *quiet storm*, que también comparten raíces históricas [musicforwardfoundation.com, 2021] y presentan apariciones simultáneas en los conjuntos de datos (tal no es el caso de los subgéneros de rock de acuerdo con la matriz de coocurrencias, pero de todas maneras en ambos casos se podría decir que hay relaciones dentro de ambos grupos de géneros).

  Cerrando este comentario, hay que aclarar que lo mencionado es solo una hipótesis y el hecho de que en géneros “únicos” (sin subgéneros o variantes) como *filmi* (música de la india) tampoco se haya logrado un *f1-score* aceptable en ambos modelos podría contradecir lo dicho.

Para cerrar esta sección, se puede concluir que los géneros en los que se alcanzaron mejores resultados fueron aquellos que poseen patrones musicales estandarizados y poco diversifi-

cados. Las canciones de música clásica, *death metal* y reggeaton tienden a seguir las mismas normas instrumentales y rítmicas correspondientes al género. Por otro lado, los géneros más difíciles de reconocer fueron los más ambiguos (como el pop rock o el rock chileno) y los que más subgéneros compartían en la lista. Otra de las características mencionadas en la disertación de Enric Gauss [Gauss, 2009] de un conjunto de canciones que se debe tener en cuenta para facilitar la tarea de la clasificación de géneros es la especificidad de los mismos géneros. *A priori*, las taxonomías generales son más fáciles de identificar que las específicas. Por ejemplo, es más fácil discernir entre *rock*, *jazz* y música clásica que entre *rock*, *folk* y *pop* pues los géneros del segundo conjunto mencionado se parecen más entre sí que los del primero. Siguiendo este hilo, al existir en la lista de los 33 géneros subconjuntos de géneros similares, no es difícil ver porqué las métricas fueron más bajas para los géneros pertenecientes a estos subconjuntos.

4.2. Matriz de confusión

Con el fin de averiguar que géneros se confunden entre sí, se presentará una matriz de confusión con respecto a los resultados de la evaluación del modelo complejo. Se eligió este modelo pues mostró un mejor desempeño que el modelo del comité.

Normalmente solo se puede realizar una matriz de confusión de más de dos clases (osea, que no sea binaria) cuando se trabaja con un problema multi-clase. En ese caso se quiere visualizar la confusión entre cada par de géneros, por lo que es necesario adoptar ciertas suposiciones para adaptar este problema multi-etiqueta al enfoque multi-clase y así poder armar una matriz de confusión.

A cada canción t del conjunto de pruebas D_{test} le corresponde una "lista de géneros" o G_t. Estos corresponden a los géneros cuyos valores correspondientes en el vector binario y_t son iguales a 1. A su vez, tomando el vector de probabilidades y'_t generado por el modelo se puede armar una lista de "géneros predecidos" o G'_t que corresponden a los géneros cuyas probabilidades en y'_t fueron mayores a 0.5. Con todo esto se define al "género con mayor probabilidad de la canción" o g^t_{max}, como el género musical que maximiza su probabilidad en y'_t. En otras palabras, g^t_{max} es el género que más probabilidades tiene de pertenecer a t según el modelo (independiente si esta probabilidad es mayor o menor a 0.5). Gracias a este valor es posible trabajar con un enfoque multi-clase, pues del vector y'_t entregado por el modelo se puede tomar g_{max}, obteniendo un valor de salida unidimensional en lugar de un vector.

Con esta información se puede armar una matriz de confusión C, inicializada como una matriz nula, realizando el siguiente procedimiento para cada género g de la lista de géneros G_t de cada canción t:

- Si $g \in G'_t$ o si $g = g^t_{max}$, se considera que el modelo identificó correctamente a g en t.

Por lo tanto: $C[g][g] \rightarrow C[g][g] + 1$

- En el caso contrario se considera que el modelo "confundió" a g por g_{max}, pues en lugar de identificar correctamente a g, realizó una predicción errónea con mayor probabilidad en g_{max}.

  Por lo tanto: $C[g][g_{max}] \rightarrow C[g][g_{max}] + 1$

Considerando que la cantidad de canciones por género no está balanceada en D_{test}, cada fila de la matriz de confusión C se debe normalizar para que se pueda realizar un mejor análisis comparativo entre filas. El resultado se muestra en la figura 22.

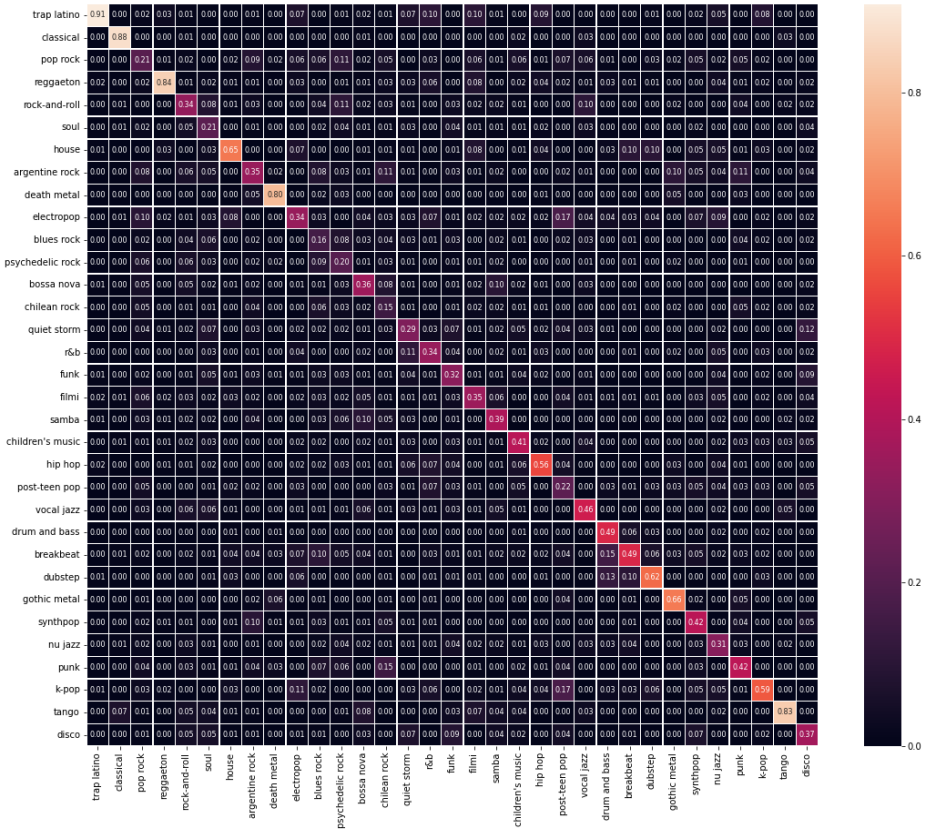


Figura 22: matriz de confusión para los resultados de la evaluación del modelo complejo en D_{test}

En términos generales se observa que para cada género las instancias en las que ocurrieron identificaciones correctas fueron mayoritarias, pues los valores en la diagonal de la matriz

fueron las mayores en cada fila. Estos valores no fueron muy altos en la mayor parte de los casos, pero considerando la gran cantidad de géneros con los que se está trabajando, es un logro. Si el modelo no hubiese aprendido ninguna de las características musicales correspondientes a cada género, cada valor de cada fila hubiese tenido estadísticamente un índice de reconocimiento cercano a $1/33 = 0{,}03$ (que es el promedio esperado considerando la cantidad de géneros). Afortunadamente esto no pasa, lo que se comprueba observando la matriz de confusión y su diagonal.

En términos de géneros específicos, es posible remarcar algunas confusiones y casos curiosos:

- Existen instancias en las que canciones de *pop rock* se interpretan erróneamente como canciones pertenecientes a algunos subgéneros de rock y pop, lo cual es de esperarse tan solo considerando al nombre de este género. *Argentine rock*, *psychedelic rock* y *post-teen pop* son algunos de los géneros que mayor índice de reconocimiento poseen en la fila del *pop rock*.

  También hay confusión con *vocal jazz* y *children's music*. En el caso del *vocal jazz* esto puede deberse a que los estilos de canto de este género son similares a los presentes en el *pop rock*. En cuanto a *children's music*, puede deberse a que las canciones incluidas en películas para niños (como las de Disney) suelen presentar características musicales que apelan al público general o lo *"mainstream"*, cosa es recurrente con el *pop rock* [MasterClass, 2022].

- Curiosamente, el grado de identificaciones erróneas en cuanto a *trap latino* en canciones de *reggaeton* y vice-versa no es muy grande, considerando lo relacionados que están estos dos géneros según la matriz de coocurrencias (figura 16). Es decir, el modelo no solo fue capaz de identificar estos dos géneros musicales en conjunto, sino que también supo reconocer de manera frecuente casos en los que solo uno de estos dos géneros hizo aparición sin el otro.

- Existen instancias en las que canciones de *rock-and-roll* fueron interpretadas como *psychedelic rock*, *soul* o *vocal jazz*. Estos tres géneros compartieron sus épocas de auge con el *rock and roll* y también se hicieron populares en estados unidos y occidente en general ([Kot, 2020], [O'Brien, 2015], [Ritz, 2021], [Bahl, 2015]). No es de extrañar entonces que estos géneros puedan compartir también características musicales y por lo tanto sean confundidos por los modelos.

- Existen instancias en las que canciones de *house* se reconocen como *Breakbeat* o *dubstep*. Esto sucede pues estos tres géneros musicales son subgéneros de la música electrónica. A su vez, algunas canciones de *dubstep* se reconocen como *breakbeat* y *drum and bass*, probablemente por razones similares.

- Existen instancias en las que canciones de rock argentino se reconocen como *pop rock*, *gothic metal*, *punk* e incluso con *rock chileno*. Todos de alguna manera son subgéneros del rock que pudieron inspirar a los estilos de muchas bandas de rock argentino (o por

lo menos las bandas cuya música se incluyó en el conjunto de canciones utilizado para este trabajo).

- Existe cierto nivel de confusión con *gothic metal* en canciones de *death metal* y viceversa, pero no es muy grande. Pasa lo mismo con el *trap latino* y el *reggaeton*: aparentemente el modelo sabe distinguir con cierta precisión cuando una canción pertenece a un género y no al otro.
- Existen instancias en las que canciones de *electropop* se identifican como *post-teen pop*, aunque esto no pasa al revés. Ambos son subgéneros del pop que apuntan a públicos juveniles [Johnson, 2016].
- Existen instancias en las que canciones de *bossa nova* se confunden con canciones de *samba*. Como se mencionó en la sección 4.1.5, estos dos géneros están muy relacionados histórica y geográficamente.
- Existen instancias en las que canciones de *k-pop* se identifican como *post-teen pop* o como *electropop*. Esto tiene sentido, pues el *k-pop*, tal como indica su nombre, es otra variante de la música pop caracterizada por su país de origen, Corea.
- *Quiet Sorm*, *r&b*, *funk* y *disco* fueron géneros musicales imperantes dentro de la cultura afroamericana entre la década de los 50's y los 70's ([musicforwardfoundation.com, 2021], [Harvey, 2012]) y por ende están relacionados, hecho que se hace notar observando la matriz de coocurrencias (figura 16). En la matriz de confusión se hacen notar algunos casos de, valga la redundancia, confusiones entre estos géneros. Algunas canciones de *R&b* se interpretan como *quiet storm*, *disco* como *quiet storm* o *funk*, *quiet storm* como *funk* o *disco* y *funk* como *disco*.

Como se puede ver en estos casos, una gran parte de las *confusiones* provocadas por parte del modelo complejo suceden entre géneros similares que muchas veces comparten historia y estilos. Por lo tanto, estas confusiones no necesariamente se deben interpretar como una demostración de la baja capacidad del modelo de discernir entre los distintos estilos musicales sino que se puede ver como una clara evidencia de cómo estos géneros se entrelazan, se inspiran y se retroalimentan entre sí.

4.3. visualización de mapas de características

Con el fin de entender el comportamiento de las redes neuronales convolucionales durante la tarea de reconocer algunos géneros en específico, se visualizan a continuación distintos mapas de características obtenidos desde el modelo del comité. Se eligió este modelo pues si bien no alcanzó resultados tan favorables como el modelo complejo, es un modelo compuesto de subredes especializadas para cada género, lo que simplifica el análisis específico a los géneros.

Los géneros escogidos para este análisis son tres de los géneros que se lograron reconocer de mejor forma (mayor*f1-score*) y los tres géneros que más costaron reconocer (menor *f1-score*) por parte del modelo del comité, según la figura 3. Estos géneros son:

- Top 3 “mejores géneros”:
  - *classical* (*f1-score*: 0.774)
  - *trap latino* (*f1-score*: 0.441)
  - *dubstep* (*f1-score*: 0.423)

  (nota: no se incluyó el *trap latino* a pesar de alcanzar un *f1-score* alto por compartir muchas similitudes con el *reggaeton*)
- Top 3 “peores géneros”:
  - *post-teen pop* (*f1-score*: 0.071)
  - *psychedelic rock* (*f1-score*: 0.127)
  - *synthpop* (*f1-score*: 0.135)

Para cada género g del grupo de “mejores géneros” se escoge para analizar la canción de D_{test} identificada correctamente como parte de g con mayor probabilidad o p_g.

Para cada género g del grupo de “peores géneros” se escogerán dos canciones: la primera será la canción de D_{test} que, si bien no pertenece a g, arrojó un mayor p_g; mientras que la segunda canción será la que, a pesar de si pertenecer a g, arrojó un menor p_g. La razón de porque se eligen estas dos canciones por cada “peor género” es porque representan los dos casos en los que el modelo se equivoca: cuando reconoce incorrectamente a un ejemplo como parte de una clase (falso positivo) y cuando reconoce incorrectamente al ejemplo como si no fuese parte de la clase (falso negativo).

Cada canción escogida “pasará” por el “miembro del comité” correspondiente al género desde el que se seleccionó la canción. De ahí, se mostrarán mapas de activación obtenidos desde cada primera capa de cada una de las 4 redes neuronales convolucionales (pertenecientes a las 4 fuentes musicales) que forman al “miembro del comité”.

A continuación, en cada figura (correspondiente a una canción) y para cada fuente musical se muestran 4 imágenes:

- espectrogramas originales: no son mapas de activación, sino los espectrogramas de las fuentes musicales que componen a la canción
- promedios de activaciones: recordando que la primera capa de cada una de las cuatro redes que forman a un “miembro del comité” posee 16 convoluciones (ver sección 3.6.2), esta imagen representa el promedio de mapas de activaciones en todas las convoluciones

- activaciones medianas: si para cada mapa de activación obtenido desde una de las 16 convoluciones se calcula la suma de sus activaciones (es decir, la suma de sus "píxeles") y se ordenan los mapas de activaciones según esta suma, esta imagen representa al mapa de activación que resultó ser la mediana.
- activaciones máximas: si para cada mapa de activación obtenido desde una de las 16 convoluciones se calcula la suma de sus activaciones (es decir, la suma de sus "píxeles"), esta imagen muestra el mapa de activación que alcanzó una mayor suma. En otras palabras, es el mapa de activación más excitado.

Las figuras 23, 24 y 25 visualizan este proceso en las canciones que fueron identificadas correctamente con mayor probabilidad para los géneros que mejor se reconocieron. Estas canciones son:

- *Dmitri Shostakovich - Suite from The Gadfly, Op. 97a: II. Contradance*: reconocida correctamente como *classical* con una probabilidad de 0.999
- *Anuel AA - Hipócrita*: reconocida correctamente como *reggaeton* con una probabilidad de 0.994
- *Virtual Riot - GOAT*: reconocida correctamente como *dubstep* con una probabilidad de 0.993

En estos casos tanto las imágenes de los promedios de activaciones como las activaciones máximas parecen ser funciones identidades de los espectrogramas originales. En cuanto a las activaciones medianas, la mayoría poseen un bajo nivel de excitación, lo que indica que para cada "miembro del comité" por lo menos un 50 % de las convoluciones no se activan mucho y solo una minoría de las convoluciones captan una porción considerable del audio original de la fuente musical (de todos modos este patrón se repetirá para la mayoría de figuras de acá en adelante).

Tal como se puede ver en la figura 23, la canción de Dmitri Shostakovich, perteneciente a música clásica, no contiene ni bajos, ni percusiones ni voz. Solo instrumentos de viento (habiendo escuchado el extracto original de la canción se sabe esto). Es posible que una de las razones de porqué la música clásica fue tan reconocible por ambos modelos es que suele presentar una característica falta de voces, bajos y (a veces) voz, lo que en combinación a sus instrumentos típicos, lo hacen un género muy reconocible.

La canción de Anuel AA, siendo una canción de reggaeton, presenta el típico ritmo del reggeaton, instrumentalizaciones sintéticas y una voz con *autotune*. Como se puede ver el la figura 24, todos estos elementos contribuyen al reconocimiento de esta canción como parte de este género.

Por último está la canción de Virtual Riot, perteneciente al género *dubstep*. Este género se caracteriza por usar lineas de bajos extendidas y manipuladas o "*wobble bass*" (aunque esto

se puede escuchar más en la separación de percusión que en la del bajo), ritmos de tipo "*snare*", tempos rápidos y sintetizadores frecuentes que lo diferencian de otros géneros de música electrónica [Montemayor, 2018]. También se usa mucho el recurso del "*bass drop*", que son cambios súbitos en el ritmo y los bajos que dan una sensación energética (aunque este recurso generalmente se ocupa en gran parte de la música electrónica) [Young, 2010]. Como se puede ver en la figura 25, las percusiones y los "otros" presentan una activación muy alta. También se puede notar el *bass drop* en la forma en la que el ritmo acelera hasta detenerse por un corto periodo para luego "explotar".

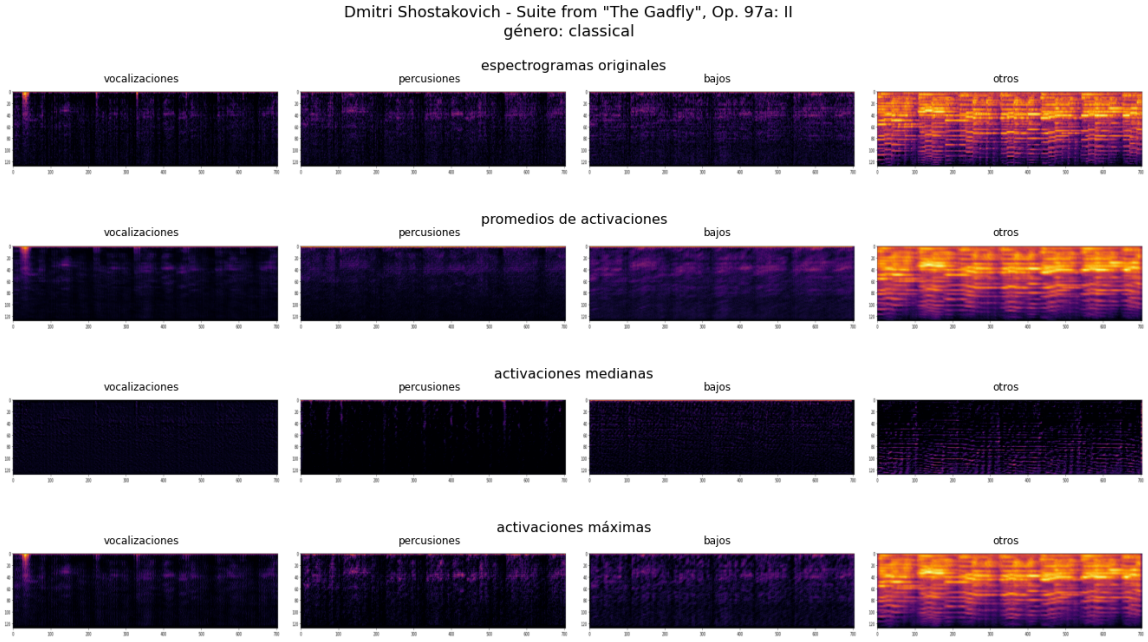


Figura 23

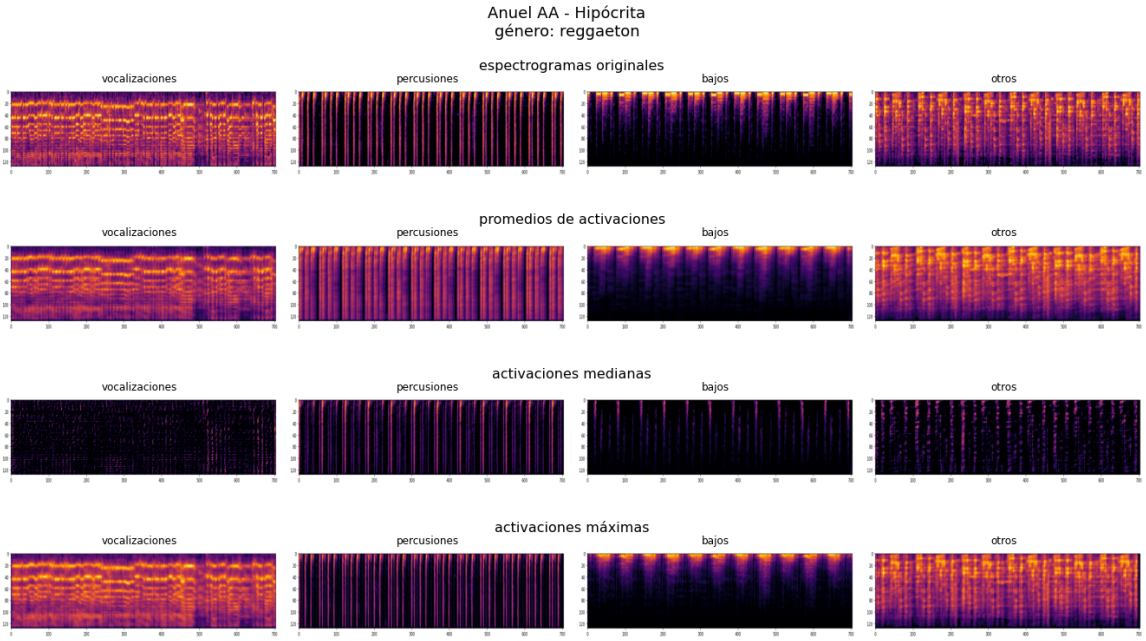


Figura 24

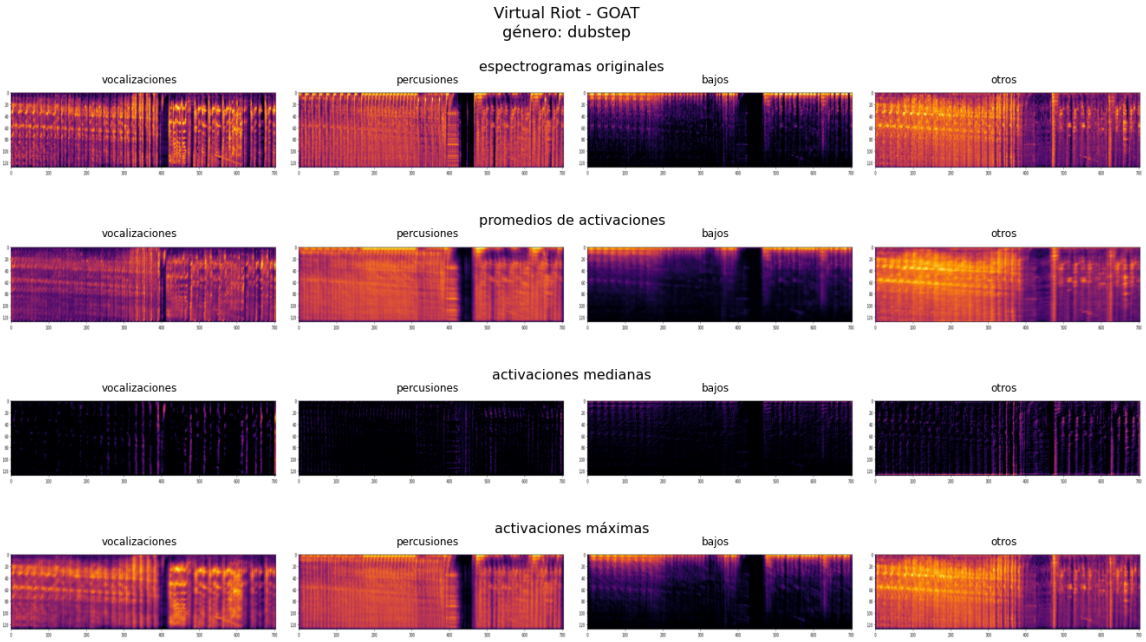


Figura 25

Las figuras 26, 27 y 28 visualizan este proceso en las canciones pertenecientes a los “peores” géneros pero en las que el modelo retornó menores probabilidades p_g. Estas canciones son:

- *Charli XCX - After the Afterparty*: a pesar de pertenecer al género *post-teen pop*, no se reconoció como tal, obteniendo una probabilidad de 0.227 de pertenecer a este género.
- *Clan of Xymox - The Same Dream*: a pesar de pertenecer al género *synthpop*, no se reconoció como tal, obteniendo una probabilidad de 0.052 de pertenecer a este género.
- *Funkadelic - Ain't That Funkin' Kinda Hard on You? - We Ain't Neva Gonna Stop Remix*: a pesar de pertenecer al género *psychedelic rock*, no se reconoció como tal, obteniendo una probabilidad de 0.022 de pertenecer a este género.

Solo viendo las figuras 26, 27 y 28 es difícil ver el porqué estas canciones no fueron reconocidas como sus respectivos géneros, aunque hay algunas diferencias menores en relación a los mapas de activación visualizados en el caso de los "mejores géneros". Hay que tener en cuenta que solo se están tomando en cuenta las activaciones de las primeras capas de cada red y que cada fuente musical pasa por otras 3 capas convolucionales (con un total de 80 convoluciones restantes) que no se están tomando en cuenta.

"After the Afterparty" de *Charli XCX* pertenece al género *post-teen pop*, que es música pop orientada a adolecentes (como su nombre indica). Debido a que lo que caracteriza a este género es su público objetivo y no su estilo musical, no es difícil entender porqué el reconocimiento fue tan deficiente por parte de este modelo. Mucha música "pop" puede o no considerarse como *post-teen pop*,lo que dificulta la tarea de identificar a las canciones que pertenecen a este género y a las que no. A parte de eso, no se puede decir mucho acerca de los mapas de activación visibles en la figura 26, a parte de que prácticamente toda la canción se recibe en las primeras capas. También es curioso que en la mediana de las vocalizaciones se activó todo el "silencio" y no se detectó activaciones en la parte cantada (siendo esto una activación "negativa").

Con *"The Same Dream"* de *Clan of Xymox*, cuyas activaciones son visibles en la figura 27, pasa algo curioso: las activaciones promedio en "otros" son muy bajas, a pesar de que las activaciones máximas sigan cumpliendo el patrón de ser casi una función de identidad. Si se considera además el hecho de que las activaciones medianas son casi nulas, se puede concluir que muy pocas de las convoluciones pudieron procesar el contenido de la canción en cuanto a instrumentalización. Este hecho pudo haber afectado directamente al reconocimiento, pues el *synthpop* se caracteriza por sus sintetizadores (tal como su nombre indica), aunque de todos modos es sólo una hipótesis. Además hay que considerar que este es uno de los muchos subgéneros de música electrónica elegidos para este trabajo, por lo que es de esperar que el reconocimiento es este género sea deficiente de acuerdo con lo mencionado en la sección 4.1.5.

Por último, *"Ain't That Funkin' Kinda Hard on You? - We Aint Neva Gonna Stop Remix"* de *Funkadelic* debería pertenecer al género de "psychedelic rock", que es un subgénero del rock nacido en la década de los 60 que intenta representar estados de alucinaciones y drogas usando guitarras sobrecargadas [O'Brien, 2015]. Sin embargo esta canción es en realidad

un *remix* en el que participaron los raperos Kendrick Lamar y *Ice Cube* [Strauss, 2016]. Al escuchar esta versión es fácil darse cuenta que parece más una canción de *hip hop* o *funk* que de *psychedelic rock*. De hecho, esta canción apenas tiene guitarras. Por lo tanto, este es uno de los casos en los que el género registrado en un artista se registra también en todas sus canciones, incluyendo a las que se desvíen de ese género como los *remixes*. Si bien *Funkadelic* aparece como un grupo de “psychedelic rock” en la API de Spotify, no todas sus canciones poseen las características de este género. Con respecto a los mapas de activación visibles en la figura 28, si bien no se puede realizar un análisis profundo a las convoluciones de la primera capa, se puede formular la hipótesis de que en las capas siguientes del comité de *psychedelic rock* las activaciones serán menores, pues esta canción no posee los elementos instrumentales que caracterizan a de este género.

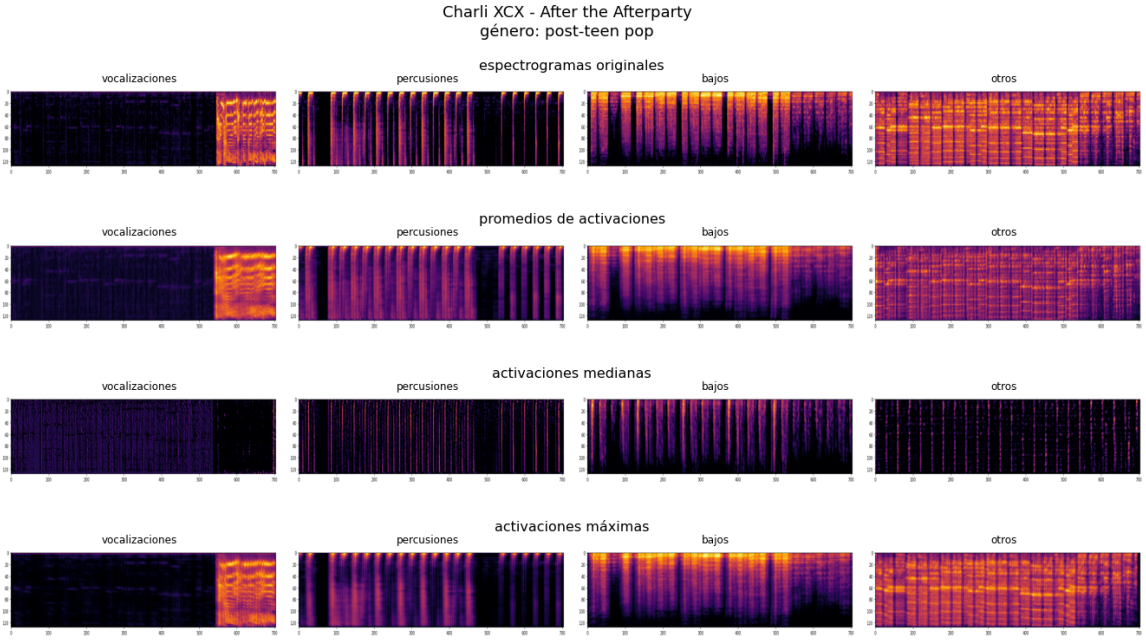


Figura 26

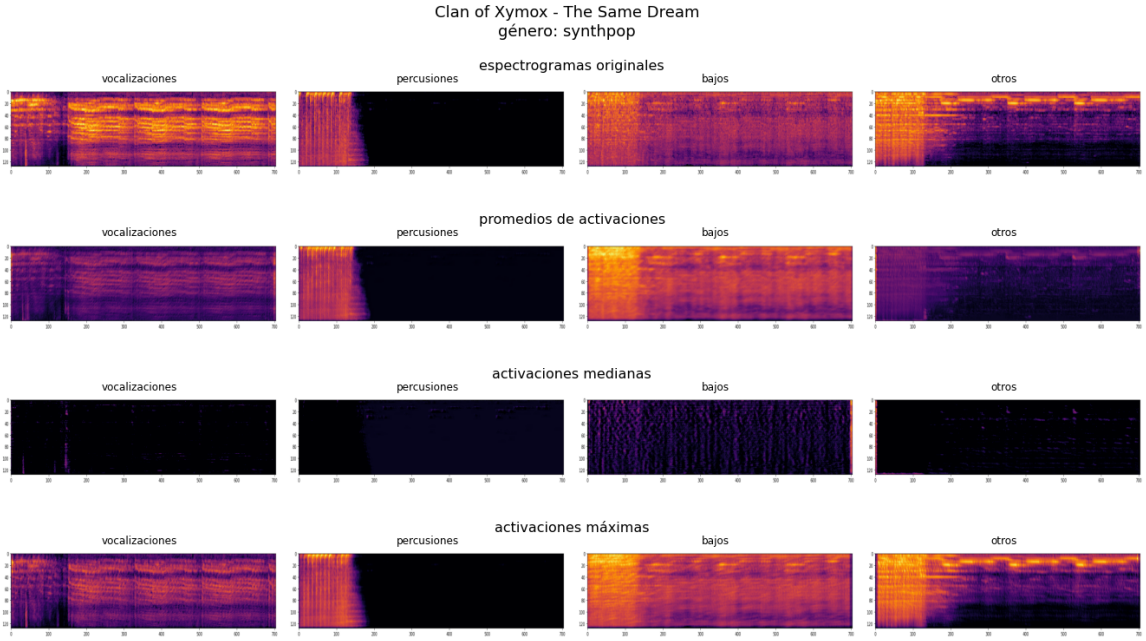


Figura 27

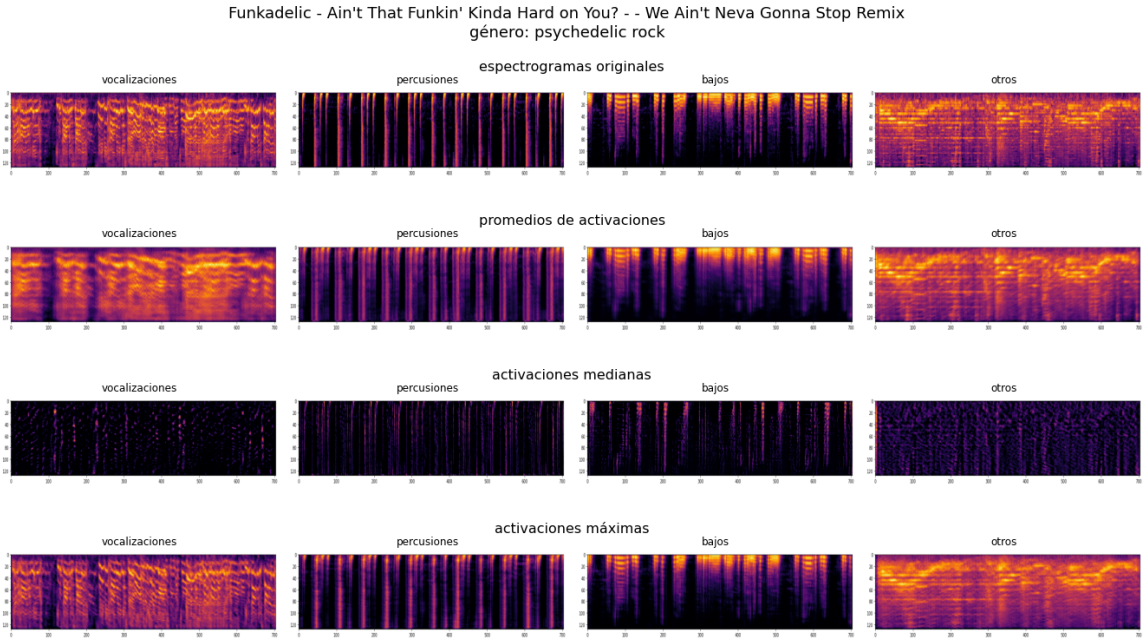


Figura 28

Las figuras 29, 30 y 31 visualizan este proceso en las canciones que, si bien no pertenecen a los “peores” géneros, el modelo arrojó mayores probabilidades p_g. Estas canciones son:

- *Dyonoro - Me Provocas*: a pesar de no pertenecer al género *post-teen pop*, si se reconoció como tal con una probabilidad de 0.961
- *The Knife - Bird*: a pesar de no pertenecer al género *synthpop*, si se reconoció como tal con una probabilidad de 0.936
- *Therion - Quetzalcoatl*: pesar de no pertenecer al género *psychedelic rock*, si se reconoció como tal con una probabilidad de 0.924

La canción *Me Provocas* de *Dyonoro* fue reconocida como *post-teen pop* con una alta probabilidad a pesar de no ser catalogada como tal. Observando la figura 29 y comparandola con la figura 26, es probable que la razón de porqué la canción *After the Afterparty* de *Charli XCX* no fue identificada como *post-teen pop* mientras que esta sí radique en la poca presencia de vocalización presente en *After the Afterparty*, pues se escogió un segmento de 15 segundos de los cuales recién al final se empieza a cantar. *Me Provocas*, por otro lado, presenta una voz femenina (lo que se sabe al escuchar la canción) en prácticamente todos los 15 segundos que dura el segmento. No es descabellado pensar entonces que la presencia del canto sea clave en la identificación de *post-teen pop*. Sin embargo, sigue siendo un género bastante ambiguo, hecho que se evidencia observando el bajo desempeño que ambos modelos obtuvieron en su identificación (ver sección 4.1.2).

Bird de *The Knife* fue reconocida como “synthpop” con una probabilidad alta sin ser parte de este género. Escuchando la canción y observando la parte “otros” de la figura 30 se puede detectar la presencia de sintetizadores minimalistas en todo el segmento (ver los patrones en la parte superior del espectrograma original de “otros”), lo que fácilmente pudo contribuir a que esta canción sea reconocida como “synthpop”.

Por último, *Quetzalcoatl* de *Therion* es una canción que fue identificada como rock psicodélico con alta probabilidad a pesar de no pertenecer a este género. Al escuchar el extracto se puede concluir que esta canción suena más a metal o metal gótico. Observando la figura 31, se detecta una actividad alta en las instrumentalizaciones u “otros”, tanto en el espectrograma original como en los mapas de activación. Es posible entonces que *Quetzalcoatl* haya sido reconocida como rock psicodélico debido a que las progresiones de guitarra pertenecientes a esta canción suenan a *psychedelic rock* según el miembro del comité designado a este género. También es curioso como la mediana de la activación de las vocalizaciones es prácticamente inversa al espectrograma original, aunque no se pueden sacar muchas conclusiones con respecto a este hecho.

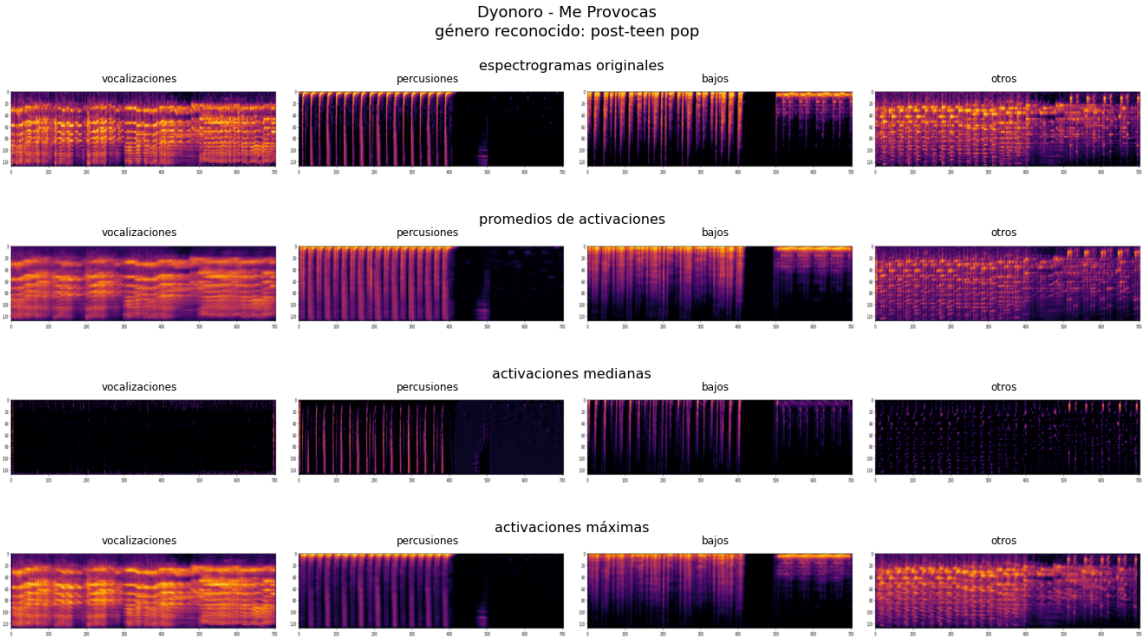


Figura 29

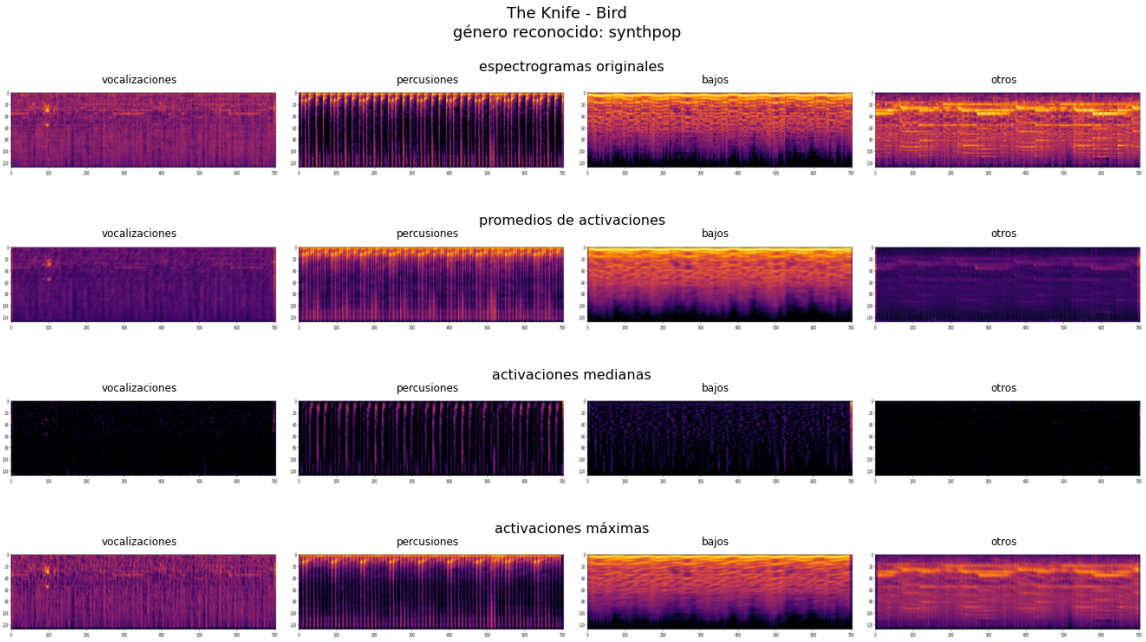


Figura 30

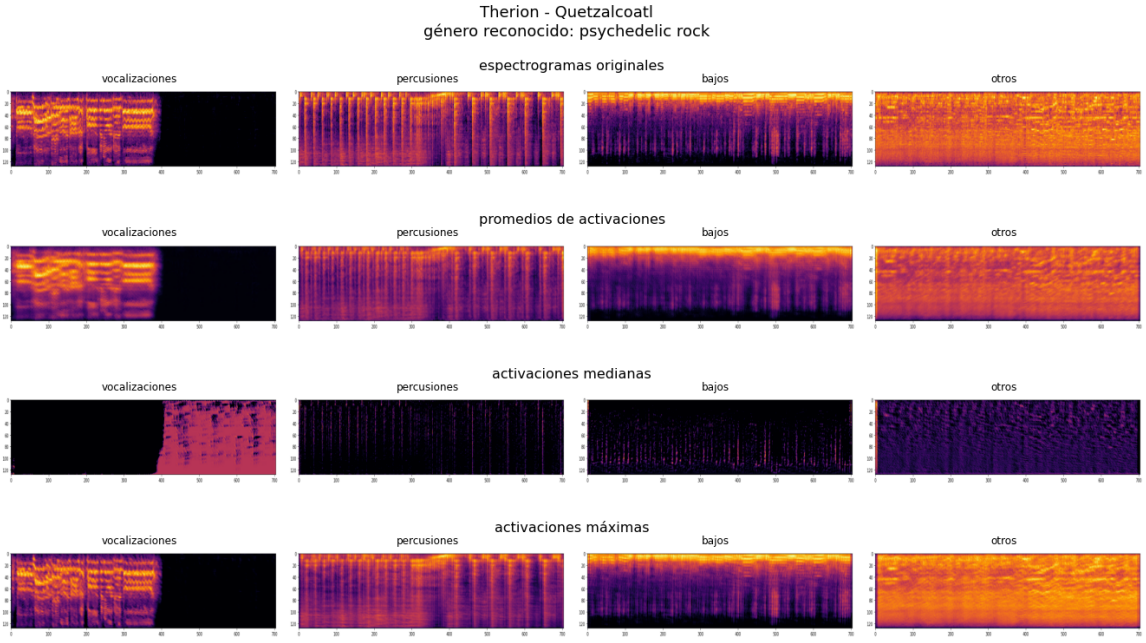


Figura 31

Para cerrar esta sub-sección, sería bueno remarcar una vez más que los mapas de activación vistos corresponden a información entregada sólo por las primeras capas de redes convolucionales de 4 capas, con un total de 96 convoluciones por red. Si bien es posible ver algunos de los elementos sonoros que más atención recibieron en la "entrada" por parte del modelo, sólo con esa información no se puede descifrar el funcionamiento completo detrás de las predicciones hechas ni por los "miembros del comité" ni, por extensión, por ambos modelos desarrollados en este trabajo. El funcionamiento de las redes neuronales es mucho más complejo de lo que se puede ver usando solo mapas de activación de las capas de entrada e incluso si se tomaran todos los mapas de activación de todas las capas de cada red, se tendría que probar con muchos más ejemplos de canciones para poder sacar observaciones más acertadas. De todos modos, en esta sección se pudieron observar varias características sonoras propias de algunos de los géneros trabajados.

CAPÍTULO 5
CONCLUSIONES

5.1. Conclusiones

El problema de la clasificación automática de géneros musicales es un tema relevante cuando se habla de la industria del *streaming* musical y los algoritmos de recomendación. Esto se aplica más aún en el caso multi-etiqueta, considerando la gigantesca cantidad de géneros musicales que se pueden encontrar en estos servicios y que suelen entrelazarse en las canciones. Sería útil entonces contar con un algoritmo que pueda predecir con un alto desempeño a qué géneros musicales pertenecen las canciones de una biblioteca de gran tamaño, para así poder formar una taxonomía universal que no dependa de información ingresada por seres humanos, más allá de las canciones en sí. Sin embargo, esta misma información es la que permite que los algoritmos de predicción que se buscan desarrollar mejoren (o por lo menos en el caso de aprendizaje supervisado) y, al ser creada por humanos, es propensa a errores y contradicciones.

El arte es derivativo, lo que quiere decir que cada pieza de arte está inspirada en mayor o menor medida por una o varias piezas de arte anteriores. Esto es parte de lo que se conoce como la "cultura del remix" [Ferguson, 2015] y se hace notar especialmente en la música. Prácticamente todo estilo musical contemporáneo es una variante o combinación de otros estilos musicales. El "problema" ocurre cuando nacen tantas variantes de un mismo estilo musical que se llega a un punto en el que ni siquiera los artistas o las personas adeptas a estas variantes son capaces de llegar a un consenso en cuanto a qué las diferencian. Si no existe una clara definición humana de las características que definen a cada "subgénero de un subgénero musical", ¿puede entonces una máquina discernir estas características o inventarse sus propias definiciones?.

Siguiendo con esta linea de pensamiento, debido a la gran cantidad de variaciones de rock y pop incluidas en este experimento, estas mismas variaciones fueron más difíciles de distinguir al ser más "vagas" en cuanto a lo que las definen. No hay duda de que si, por ejemplo, todas las variantes de *rock* utilizadas para este trabajo como *blues rock* o *psychedelic rock* se hubiesen combinado en una única categoría "*rock*", la identificación hubiese sido más exitosa. Esto también puede decirse para todas las subcategorías de música electrónica con las que se trabajó como *dubstep* o *electropop*. Por otro lado, al generalizar esta taxonomía se estarían traicionando las motivaciones detrás de este trabajo pues descartar un subgénero musical podría ser equivalente a ignorar su relevancia. Por lo tanto, es necesario encontrar un equilibrio en cuanto a la taxonomía de los géneros musicales escogidos para trabajos futuros, que incluya estilos musicales diversos pero a la vez no muy vagos.

El rock chileno y el rock argentino son dos géneros que se incluyeron para este trabajo con fines experimentativos. La idea era verificar si una red neuronal podría solo con el audio de

las canciones distinguir a estos dos géneros que, además de ser *rock*, solo se definen por sus países de origen. Como se observó en los resultados, los modelos no aprendieron a hacer una distinción clara de estos dos géneros. Es posible además que la mera inclusión del rock chileno y rock argentino haya dificultado al reconocimiento por parte de los modelos de los otros subgéneros del *rock* incluidos en este trabajo por lo mencionado en el párrafo anterior.

Los géneros musicales que se lograron predecir con mayor exactitud fueron aquellos con características más generales y contenían características sonoras extremadamente distinguibles, como es el caso de la música clásica, el tango, el *reggeaton*, el *trap latino*, el *death metal* y el *gothic metal*. A pesar de que tanto el *reggeaton* y el *trap latino* como el textitdeath metal y el *gothic metal* son pares de géneros relacionados entre sí, estos estilos musicales siguen poseyendo elementos muy identificables para los modelos utilizados.

En términos más específicos al experimento y a los modelos entrenados, el enfoque *committee machine* no logró resultados tan buenos en comparación con el modelo complejo. La razón principal de porqué sucedió esto es la falta de recursos. Los conjuntos de entrenamiento utilizados para entrenar a los "miembros del comité" fueron demasiado pequeños para esta tarea y no fueron lo suficientemente representativos en cuanto a los casos en los que el género correspondiente a cada miembro del comité no se hacía presente. El modelo complejo logró mejores resultados, pero la mayoría de géneros musicales demostraron ser difíciles de identificar.

Otro factor que podría mejorar los resultados es la cantidad de recursos y ejemplos con los que se trabaja. Es muy posible que utilizar conjuntos de canciones más grandes como el *MUMU dataset* [Oramas *et al.*, 2017] mejoraría los resultados. Desgraciadamente este trabajo fue limitado a los recursos que *Google Colab pro* ofrecía.

Para este trabajo se escogió a la biblioteca de *Spotify* como la fuente de datos desde la cual se obtuvo el conjunto de canciones utilizado. Esta elección se realizó tomando en cuenta varias características que le dan ventaja a esta plataforma: el vasto número de canciones disponibles, la diversa variedad de géneros presentes y el hecho de poder encontrar múltiples géneros indicados para cada canción. El problema es que estos géneros no dependen de la canción ni del álbum, sino del artista. Todos los géneros indicados para un artista se indican también en todas sus canciones. Esto trae muchos problemas, pues se asume que cada artista no puede variar ni experimentar con su música durante su carrera musical. Por ejemplo, si un artista de *punk* decide alejarse de estilo y hacer canciones de *pop rock*, esas nuevas canciones se considerarán como *punk* aunque no lo sean. Como es de imaginarse, esto impactó negativamente al desarrollo de los modelos.

Como parte del procesamiento del audio utilizado para los procesos de entrenamiento y pruebas de los modelos se utilizó *demucs*, una herramienta también basada en modelos de aprendizaje capaz de separar a cada canción en cuatro fuentes musicales. Se escogió *demucs* pues analizar una canción separada por vocalización, bajos, percusiones y otros facilita de manera potencial el análisis de la canción, especialmente cuando se intenta predecir a qué estilos musicales pertenece. Con el fin de verificar el impacto de esta separación en el

desempeño de los modelos, sería útil repetir todo este trabajo sin la separación de fuentes musicales (es decir, alimentando a los modelos usando el audio original de las canciones) y comparar los resultados.

5.2. Trabajo a futuro

El experimento realizado da pie para futuros estudios, de los cuales se proponen los siguientes:

- Repetir el experimento, pero sin separar las fuentes musicales. De este modo se podría realizar una comparativa en cuánto aporta esta separación (si es que aporta). También se podría mantener la separación, pero utilizar como entrada sonido en formato *waveform* en lugar de espectrogramas.
- Repetir el experimento, pero utilizando más datos por canción aparte del sonido como el nombre del artista, el año de lanzamiento o el país de origen del artista. Este último valor podría facilitar el reconocimiento de géneros específicos a países, como el rock chileno o el rock argentino. Sergio Oramas, Oriol Nieto, Francesco Barbieri y Xavier Serra [Oramas *et al.*, 2017] utilizaron, además del sonido de las canciones, datos como las portadas de los álbumes o las letras de las canciones para alimentar a sus modelos de clasificación. También sería útil encontrar una forma de obtener la información de a qué género pertenecen las canciones desde las mismas canciones o los álbumes y no desde los artistas.
- Repetir el experimento, pero invirtiendo más recursos como memoria RAM y un mayor tamaño de la biblioteca de canciones utilizadas, principalmente en los conjuntos de entrenamiento.
- Definir una taxonomía óptima de géneros musicales que no sea ni demasiado diversa ni demasiado general, para luego repetir el experimento con este nuevo conjunto de estilos, logrando mejores resultados. Quizás utilizar un sistema de jerarquías de géneros.
- Haciendo uso del modelo separador de fuentes musicales *demucs*, se propone el desafío de una separación más específica de fuentes. Por ejemplo, la fuente musical “otros” se podría separar aún más en instrumentos de cuerda, instrumentos de viento, pianos, etc. Con esta separación más profunda la tarea de la identificación podría facilitarse.
- Si alguna vez se logra desarrollar un modelo de clasificación de géneros musicales multi-etiqueta que demuestre un desempeño considerablemente alto, se propone también el desafío de desarrollar una red neuronal generativa que, valga la redundancia, genere canciones desde uno o varios géneros musicales de entrada.

ANEXOS

En los Anexos se incluye todo aquel material complementario que no es parte del contenido de los capítulos de la memoria, pero que permiten a un lector contar con un contenido adjunto relacionado con el tema.

REFERENCIAS BIBLIOGRÁFICAS

[Abadi *et al.*, 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., y Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Academo, 2016] Academo (2016). spectrum analizer. Recuperado de: Spectrum Analyzer | Academo.org - Free, interactive, education., fecha de ingreso: 3 de diciembre de 2021.

[Alfaro, 2021] Alfaro, R. (2021). Clasificación de textos multi-etiquetados con representación dependient, e de la etiqueta. Tésis de Pregrado. Universidad Técnica Federico Santa María, Valparaíso, Chile.

[Alpaydin, 2020] Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

[Apolo, 2022] Apolo, M. J. (2022). Transferencia de estilo bimodal desde composición musical a imagen utilizando modelos generativos profundos. Tésis de Pregrado. Universidad Técnica Federico Santa María, Santiago, Chile.

[Appleton y Perer, 1975] Appleton, J. y Perer, R. (1975). The development and practice of electronic music. *Prentice-Hall*.

[Arce, 2019] Arce, J. I. B. (2019). La matriz de confusión y sus métricas. Recuperado de: La matriz de confusión y sus métricas – Inteligencia Artificial – (juanbarrios.com).

[Bahl, 2015] Bahl, M. (2015). Vocal jazz: 1917-1950. *allaboutjazz*. Recuperado de: Vocal Jazz: 1917-1950 article @ All About Jazz.

[Barkus, 2021] Barkus, F. (2021). Streaming surpasses radio as the top way to listen to music. *CBS News*. Recuperado de: Streaming Surpasses Radio as the Top Way to Listen to Music - CBS News.

[Beardsley y Leech-Wilkinson, 2009] Beardsley, R. y Leech-Wilkinson, D. (2009). A brief history of recording to ca. 1950. *CHARM*. Recuperado de: A Brief History of Recording to ca. 1950 (rhul.ac.uk).

[Bennett, 2020] Bennett, G. (2020). The precision-recall trade-off. Recuperado de: The Precision-Recall Trade-Off. By George Bennett | by Datascience George | Medium .

[Bhatt, 2018] Bhatt, B. (2018). Micro & macro precision for imbalanced multi-class classification | machine learning. Recuperado de: (1) Micro & Macro Precision For Imbalanced Multi-class Classification | Machine Learning - YouTube.

[bnbmusiclessons.com, 2015] bnbmusiclessons.com (2015). Difference between samba and bossa nova. *B&B Music Lessons*. Recuperado de: Difference Between Samba and Bossa Nova (bnbmusiclessons.com).

[Bracewell y Bracewell, 1986] Bracewell, R. N. y Bracewell, R. N. (1986). *The Fourier transform and its applications*, volumen 31999. McGraw-hill New York.

[Brownlee, 2020] Brownlee, J. (2020). Multi-class imbalanced classification. *Machine Learning Mastery*. Recuperado de: Multi-Class Imbalanced Classification (machinelearningmastery.com).

[Capobianco *et al.*, 2021] Capobianco, G., Cerrone, C., Di Placido, A., Durand, D., Pavone, L., Russo, D. D., y Sebastiano, F. (2021). Image convolution: a linear programming approach for filters design. *Soft Computing*, 25:8941–8956.

[Castro *et al.*, 2019] Castro, W., Oblitas, J., De la Torre, M., Cotrina, C., Bazán, K., y Avila-George, H. (2019). Using machine learning techniques and different color spaces for the classification of cape gooseberry (physalis peruviana l.) fruits according to ripeness level.

[Charif *et al.*, 2010] Charif, R., Strickman, L., y Waack, A. (2010). Raven pro 1.4 user's manual. *The Cornell Lab of Ornithology Ithaca NY*.

[Charte *et al.*, 2015] Charte, F., Rivera Rivas, A., Del Jesus, M. J., y Herrera, F. (2015). Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, 163.

[Chaudhary, 2020] Chaudhary, K. (2020). Understanding audio data, fourier transform, fft and spectrogram features for a speech recognition system. *towardsdatascience*. Recuperado de: Understanding Audio data, Fourier Transform, FFT and Spectrogram features for a Speech Recognition System | by Kartik Chaudhary | Towards Data Science.

[Dai *et al.*, 2016] Dai, J., Liang, S., Xue, W., Ni, C., y Liu, W.-J. (2016). Long short-term memory recurrent neural network based segment features for music genre classification. pp. 1–5.

[Daniels y Metaxas, 2017] Daniels, Z. y Metaxas, D. (2017). Addressing imbalance in multilabel classification using structured hellinger forests. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

[Défossez *et al.*, 2019] Défossez, A., Usunier, N., Bottou, L., y Bach, F. (2019). Music source separation in the waveform domain.

[Díaz, 2018] Díaz, M. G. (2018). Cómo diferenciar el reggaetón del trap, el polémico género musical que arrasa en medio mundo. *BBC News Mundo*. Recuperado de: Cómo diferenciar el reggaetón del trap, el polémico género musical que arrasa en medio mundo.

[Ferguson, 2015] Ferguson, K. (2015). Everything is a remix. Recuperado de: Everything is a Remix.

[Gandhi *et al.*, 2017] Gandhi, A., Diner, J., Wong, R., Mani, S., y Singh, T. (2017). Understanding origins and fusion of music genres. Recuperado de: Music Genres Over the Decades (shouvikmani.github.io).

[Ganesh, 2019] Ganesh, P. (2019). Types of convolution kernels : Simplified. *towardsdatascience*. Recuperado de: Types of Convolution Kernels : Simplified | by Prakhar Ganesh | Towards Data Science.

[Gauss, 2009] Gauss, E. (2009). *Audio content processing for automatic music genre classification*. PhD dissertation, Universitat Pompeu Fabra. Departament de Tecnologies de la Informació i les Comunicacions.

[Google, 2022] Google (2022). Colaboratory - frequently asked questions. Recuperado de: Google Colab y de Colab Pro and Pro+ Sign Up Page.

[Grossi y Buscema, 2008] Grossi, E. y Buscema, M. (2008). Introduction to artificial neural networks. *European journal of gastroenterology & hepatology*, 19:1046–1054.

[Harris *et al.*, 2020] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., y Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

[Harvey, 2012] Harvey, E. (2012). The quiet storm. *pitchfork*. Recuperado de: The Quiet Storm | Pitchfork.

[Johnson, 2016] Johnson, P. (2016). Electropop target audience survey. Recuperado de: Electropop Target Audience Survey by Phoebe Johnson (prezi.com).

[Kot, 2020] Kot, G. (2020). rock and roll. *Encyclopedia Britannica*. Recuperado de: rock and roll | History, Songs, Artists, & Facts | Britannica.

[Kumar, 2020] Kumar, A. (2020). Micro-average & macro-average scoring metrics – python. Recuperado de: Micro-average & Macro-average Scoring Metrics - Python - Data Analytics (vitalflux.com) .

[Lawton, 2022] Lawton, G. (2022). What is predictive modeling? *techtarget*. Recuperado de: What is Predictive Modeling? (techtarget.com).

[Lee y Varaiya, 2003] Lee, E. y Varaiya, P. (2003). *Structure and interpretation of signals and systems*.

[Li *et al.*, 2003] Li, T., Ogihara, M., y Li, Q. (2003). A comparative study on content-based music genre classification. pp. 282–289.

[Ling y Sheng, 2010] Ling, C. X. y Sheng, V. S. (2010). *Class Imbalance Problem*, pp. 171–171. Springer US, Boston, MA.

[Log, 2021] Log, S. (2021). Precision, recall, & f1 score intuitively explained. Recuperado de: Precision, Recall, & F1 Score Intuitively Explained - YouTube.

[MasterClass, 2022] MasterClass (2022). Pop rock music guide: A brief history of pop rock. Recuperado de: Pop Rock Music Guide: A Brief History of Pop Rock - 2022 - MasterClass .

[McDonald, 2013] McDonald, G. (2013). Every noise at once. Recuperado de: Every Noise at Once.

[McFee *et al.*, 2015] McFee, B., Raffel, C., Liang, D., Ellis, D., Mcvicar, M., Battenberg, E., y Nieto, O. (2015). librosa: Audio and music signal analysis in python. pp. 18–24.

[Melville y Sindhwani, 2010] Melville, P. y Sindhwani, V. (2010). Recommender systems. En *Encyclopedia of Machine Learning*.

[Montemayor, 2018] Montemayor, M. (2018). Dubstep | todo lo que debes saber. Recuperado de: DUBSTEP | TODO lo que debes saber - Majo Montemayor.

[musicforwardfoundation.com, 2021] musicforwardfoundation.com (2021). Exploring the history of black music. *musicforwardfoundation*. Recuperado de: Exploring the History of Black Music - Music Forward Foundation.

[Navlani, 2019] Navlani, A. (2019). Neural network models in r. *datacamp*. Recuperado de: ANN (Artificial Neural Network) Models in R: Code & Examples on How to Build Your NN - DataCamp.

[O'Brien, 2015] O'Brien, L. M. (2015). psychedelic rock. *Encyclopedia Britannica*. Recuperado de: psychedelic rock | music | Britannica.

[Oramas *et al.*, 2017] Oramas, S., Nieto, O., Barbieri, F., y Serra, X. (2017). Multi-label music genre classification from audio, text, and images using deep features.

[Pendlebury, 2021] Pendlebury, T. (2021). Best music streaming service for 2021. *CNET*. Recuperado de: Best music streaming service for 2021 (cnet.com).

[radio.darkness.com, 2015] radio.darkness.com (2015). Gothic metal. Recuperado de: Gothic Metal | Radio Darkness.

[Read y Pérez-Cruz, 2015] Read, J. y Pérez-Cruz, F. (2015). Deep learning for multi-label classification. *ArXiv*, abs/1502.05988.

[Ritz, 2021] Ritz, D. (2021). Soul music. *Encyclopedia Britannica*. Recuperado de: soul music | Definition, Songs, Artists, & Facts | Britannica.

[Roberts, 2020] Roberts, L. (2020). Understanding the mel spectrogram. *Medium*. Recuperado de: Understanding the Mel Spectrogram | by Leland Roberts | Analytics Vidhya | Medium, fecha de ingreso: 3 de diciembre de 2021.

[Rodgers, 2020] Rodgers, K. (2020). Since when was escape room a genre? Recuperado de: Why There Are So Many Weird Spotify Wrapped Genres - PAPER (papermag.com).

[Santiago, 2020] Santiago, G. (2020). Samba vs bossa nova - ¿what is the difference? Recuperado de: Samba VS Bossa Nova - What's The Difference ? | Ep.28 - YouTube.

[Santos *et al.*, 2018] Santos, M., Soares, J., Henriques Abreu, P., Araujo, H., y Santos, J. (2018). Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches. *IEEE Computational Intelligence Magazine*, 13:59–76.

[Spotify, 2018] Spotify (2018). Spotify web documentation. Recuperado de: developer.spotify.com/documentation/web-api, fecha de ingreso: 1 de abril de 2022.

[Strauss, 2016] Strauss, M. (2016). Funkadelic share new .ªin't that funkin' kinda hard on you?remix with kendrick lamar and ice cube. *pitchfork*. Recuperado de: Funkadelic Share New .ªin't That Funkin' Kinda Hard on You?Remix With Kendrick Lamar and Ice Cube | Pitchfork.

[Tharwat, 2018] Tharwat, A. (2018). Classification assessment methods: a detailed tutorial.

[thepeoplehistory.com, 2017] thepeoplehistory.com (2017). Music history including genres styles, bands and artists over 90 years. Recuperado de: usic History including Genres Styles, Bands And Artists over 90 years (thepeoplehistory.com).

[Tresp, 2001] Tresp, V. (2001). *Committee machines*, pp. 1–5.

[Tsoumakas y Katakis, 2009] Tsoumakas, G. y Katakis, I. (2009). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3:1–13.

[Tzanetakis y Cook, 2002] Tzanetakis, G. y Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302.

[Veen, 2013] Veen, B. V. (2013). Short-time fourier transform and the spectogram. Recuperado de: Short-time Fourier Transform and the Spectogram - YouTube.

[Velardo, 2020] Velardo, V. (2020). Understanding audio signals for machine learning. Recuperado de: Understanding Audio Signals for Machine Learning - YouTube.

[Vera, 2011] Vera, F. (2011). La historia del tango. Recuperado de: La historia del tango - YouTube.

[Yamashita *et al.*, 2018] Yamashita, R., Nishio, M., Do, R., y Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9:611–629.

[Young, 2010] Young, R. (2010). *La guida alla musica moderna di Wire*. Isbn Edizioni.