Analizando cambios de emoción en música popular

Rodrigo Ernesto Alvarado Villeda

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Analizando cambios de emoción en música popular

Trabajo de graduación presentado por Rodrigo Ernesto Alvarado Villeda para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,

2021

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Analizando cambios de emoción en música popular

Trabajo de graduación presentado por Rodrigo Ernesto Alvarado Villeda para optar al grado académico de Licenciado en Ingeniería en Ciencias de la Computación y Tecnologías de la Información

Guatemala,

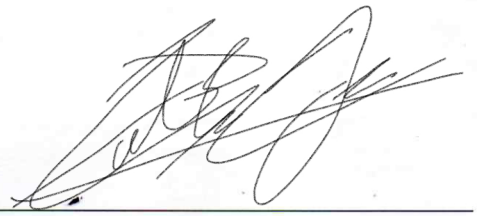
2021

Vo. Bo.:

(f) _____

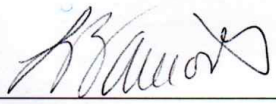
Ing. Pablo Estrada

Tribunal Examinador:



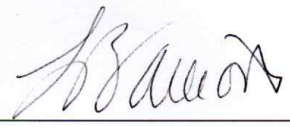
(f) _____

Ing. Pablo Estrada



(f) _____

Ing. Douglas Leonel Barrios



(f) X _____

Ing. Samuel Chávez

Fecha de aprobación: 11 junio 2021

Prefacio

Quiero dedicar el presente trabajo a quienes a lo largo de la carrera y de mi vida han tenido el mayor impacto, personas sin las cuales no me encontraría en la posición de entregar el mismo:

- A mis padres Johanna Villeda y Fredy Alvarado por su apoyo incondicional, esfuerzo para sacarme adelante y palabras de sabiduría a lo largo de los años.
- A mis primos Liby Alvarado y David Chan por aceptarme en su hogar y brindarme todo el amor y apoyo de unos padres, sin su ayuda no hubiera sido posible llegar hasta aquí
- A mi pareja Jacqueline Guarcax por motivarme a alcanzar mis metas, por su amor, y por ser una inspiración para seguir adelante.
- A mi grupo de amigos por darle color a la vida universitaria, llevarle alegría y risas a los momentos mas duros y brindarme toda la ayuda que pudieron.

Índice

Lista de figuras

Lista de cuadros

Resumen

La investigación pretende obtener un modelo de predicción de emociones para los 4 cuadrantes del modelo valencia-*arousal* con precisión de al menos 50 % utilizando *features* que han sido extraídas a partir de únicamente la música, sin tomar en cuenta las letras de la misma. Con la finalidad de lograr este objetivo se implementaron 3 algoritmos de aprendizaje de diferente índole, *SVM (Support Vector Machine)* un algoritmo de clasificación supervisado que utiliza como barrera de decisión un hiper-plano en n-dimensiones, *SVR (Support Vector Regresion)* un algoritmo de regresión basado en los mismos principios que las *SVM* y una Red neuronal implementada como algoritmo de clasificación no supervisado. Los tres modelos propuestos satisfacen este requerimiento: El *SVM* con precisión del 62.35 %, la combinación de dos *SVR* (uno entrenado para predecir valencia y el otro para predecir *arousal*) con 50.03 % y la Red neural con 61.85 %. El modelo creado utilizando *SVM* fue el que mostró mejor rendimiento en precisión además de ser el modelo menos afectado por la limitada cantidad de datos en el conjunto utilizado.

Abstract

This papers goal is to implement an emotion prediction model for the 4 quadrants of the valence-arousal scale with, at least, 50 % accuracy utilizing features extracted only from the music, not taking into account its lyrics. To achieve this goal, 3 machine learning models were implemented: A SVM (Support Vector Machine), a clasification algorithm that utilizes a n-dimentional hyperplane as its decision barrier; a SVR (Support Vector Regression), a regression algorithm based on the same principles as the SVM; and a Neural Network an unsupervised learning algorithm implemented as a clasifier. All 3 models satisfied this condition, the SVM model had an accuracy of 61.85 %, a combination of 2 SVRs (the first trained for valence and the second for arousal) had an accuracy of 50.03 % and the neural network had an accuracy of 61.85 %. The SVM model yielded the best results in accuracy and also was the least affected by the limited amount of data available.

CAPÍTULO 1

Introducción

Basado en el trabajo de análisis de emoción de canciones realizado por anteriores alumnos de la universidad, el proyecto consiste en analizar, predecir y concluir sobre los sentimientos de alegría y tristeza sobre un fragmento de canción de alrededor de 40 segundos. Con ayuda de Redes neuronales y *Support Vector Machines* como algoritmos de clasificación y aprendizaje de máquina.

El propuesto estudio busca ayudar a entender el efecto de la música en el ser humano, y está inspirado por el interés personal del autor en el estudio de la música. Sin embargo, las aplicaciones del modelo de aprendizaje de máquina propuesto son interesantes. Por ejemplo, la implementación de un sistema de recomendaciones automático que se base en el estado actual de usuario, o una guía para compositores y productores musicales que les permita transmitir lo que desean con sus piezas.

CAPÍTULO 2

Antecedentes

El campo de la detección emocional ha atraído a multitud de investigadores; computólogos, científicos de datos, psicólogos, etc. Es natural entonces, existan precedentes para la investigación de emociones en el campo de la música tan atrás como 1990.

Inicialmente, la investigación se enfocó en el uso de características extraídas de las señales de audio mismas para el entrenamiento de modelos de aprendizaje de máquina. El modelo de descripción emocional más comúnmente utilizado es el valencia-*arousal:*, que define las emociones en un espacio continuo, habitando un plano 2D, Este modelo permite mayor flexibilidad que aquellos que asignan a las piezas musicales etiquetas sentimentales.

Últimamente, el análisis sentimental se ha enfocado en el estudio de las letras, tomando ventaja de los avances surgidos del campo del *NLP*. Este enfoque queda evidenciado en los recientes trabajos de *Quantitative Sentiment Analysis of Lyrics in Popular Music* [1], *Sentiment vector space model for lyric-based song sentiment classification* [2] y *An Analysis of Music Lyrics by Measuring the Distance of Emotion and Sentiment* [3].

En el presente trabajo se buscó revistar el enfoque de los

años 90, pero aplicando técnicas de procesamiento, poder computacional y algoritmos de aprendizaje de máquina a los cuales no se tenía acceso. Esto puede, si bien no superar los algoritmos basados en letras, sentar bases para un enfoque híbrido o ser utilizado para la multitud de piezas musicales en las cuales no hay letras presentes.

CAPÍTULO 3

Justificación

El efecto sentimental que tiene la música sobre el ser humano ha sido ampliamente documentado desde la perspectiva de las ciencias psicológicas, tal y como se evidencia en trabajos como [4] y [5]; este proyecto está motivado por el deseo de analizar más a fondo qué es lo que causa estas emociones utilizando un enfoque de ciencias de la computación.

Se han realizado esfuerzos en cuanto a análisis de emoción en señales de audio. Sin embargo, estos se han enfocado, como en los siguientes casos, en el análisis a través de texto y técnicas de NLP. Ejemplos de esto existen en *Quantitative Sentiment Analysis of Lyrics in Popular Music* [1], *Sentiment vector space model for lyric-based song sentiment classification* [2] y *An Analysis of Music Lyrics by Measuring the Distance of Emotion and Sentiment* [3].

El ejemplo más moderno de uso de análisis de emoción basado en sonido fue realizado en la universidad de Cornell, por Adit Jamdar, Jessica Abraham, Karishma Khanna y Rahul Dubey en 2015 [6]. Aun así, estos se apoyaron también en el análisis lírico; se generalizó una emoción para la pieza musical entera y no utilizaron técnicas modernas de análisis de señales.

El presente proyecto aporta otra perspectiva sobre los fac-

tores que afectan la manera en la que las personas experimentan la música al enfocarse puramente en la melodía de las canciones, no en sus letras. Este enfoque puede no solamente complementar los trabajos realizados por otros investigadores (siendo la base para un sistema híbrido de clasificación) sino ser utilizado para todas aquellas canciones que los modelos basados en letras son incapaces de abarcar; estas pueden ser piezas sin letras o bien canciones en idiomas en los cuales el procesamiento de lenguaje natural no está tan refinado como en el inglés.

Las aplicaciones de tal análisis podrían beneficiar tanto a los productores musicales (dado que pueden enfocar su creación a generar una emoción deseada) como a distribuidoras de música, que pueden recomendar a sus usuarios artistas o canciones que comparten características que causan emociones .

CAPÍTULO 4

Objetivos

4.1. Objetivo general

Diseñar e implementar un modelo predictivo que indique la pertenencia de una pieza musical a uno de los cuatro cuadrantes en el modelo valencia-*arousal*, con al menos un 50 % de éxito.

4.2. Objetivos específicos

- Implementar modelos predictivos categóricos y comparar su efectividad en el análisis de audio.
- Identificar los cuadrantes con mayor facilidad de ser clasificados correctamente.

CAPÍTULO 5

Alcance

El presente proyecto no pretende comparar todos los algoritmos de aprendizaje de máquina disponibles, sino exponer ejemplos significativos y comparar sus ventajas y desventajas. Los algoritmos a utilizar son un *SVM* (Algoritmo de clasificación supervisado), una Red neuronal (Algoritmo de clasificación no supervisado) y *SVR* (algoritmo de regresión). Todo utilizando herramientas de aprendizaje de máquina disponibles de manera gratuita como *scikit-learn*, *Keras*, etc.

CAPÍTULO 6

Marco teórico

En este capítulo se introducen conceptos teóricos vitales para la comprensión y desarrollo de algoritmos de clasificación emocional.

6.1. Emoción

El concepto de emoción es uno para el cual no existe una definición consensuada y es normalmente descrito como la extensión de una lista de emociones fundamentales: enojo, miedo, tristeza, etc [7]. En el contexto de este trabajo, la emoción será definida por el modelo que la describe.

6.1.1. Modelos de clasificación de emoción

En el contexto de aprendizaje de máquina se consideran dos modelos principales, uno utiliza categorías discretas, mientras su contra-parte considera un espacio continuo. [8]

1. **Categorías discretas**: Cicero y Graver [8] organizaban emociones en 4 categorías básicas: miedo, dolor, placer y

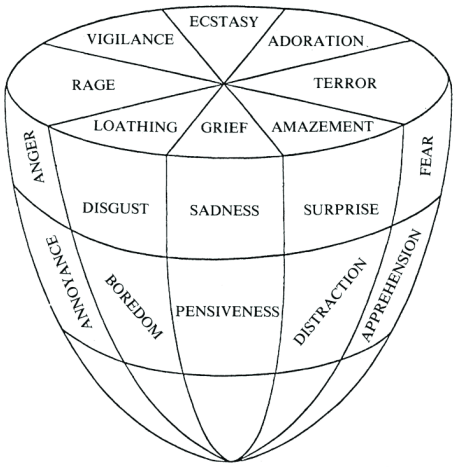


Figura 1: Rueda de emociones propuesta por Plutchik.

lujuria. Ekman describió las emociones como "discretas, medibles y fisio-relacionadas, emergiendo a partir de funciones fisiológicas y comunicativa". Mientras tanto Plutchik propuso clasificar las emociones en la forma de una rueda que incorpora 8 emociones básicas: alegría, confianza, miedo, sorpresa, tristeza, enojo, anticipación y asco [8]como se puede ver en la Figura 1. Estas emociones pueden combinarse para dar paso a unas mas complejas, además incorpora niveles de intensidad, en donde las emociones mas fuertes se ven al tope de la rueda y aquellas con menor intensidad se presentan mas cerca del fondo.

2. **Espacio continuo**: Una descripción continua debe contar con la capacidad de describir la correlación entre dos emociones (que tan cerca se encuentra una de la otra en el espacio) así como cuantificar su intensidad (por ejemplo la diferencia entre estar triste y muy triste). El primer acercamiento fue propuesto por Rosensohn en 1963, basándose en las teorías de emoción de Willhelm Wundt [9], esta clasificación utilizaba un espacio tridimensional en los ejes *pleasure-displeasure, excitement-inhibition,tension-relaxation.* Mas adelante, Schmidt *et al* sugirió un modelo bidimensional basado en valencia y *arousal*, la valencia describe que tan positiva o negativa es la emoción (placentera o no placentera) y *arousal* su nivel de intensidad [10] une ejemplo puede verse en la Figura 2. Existen otros modelos

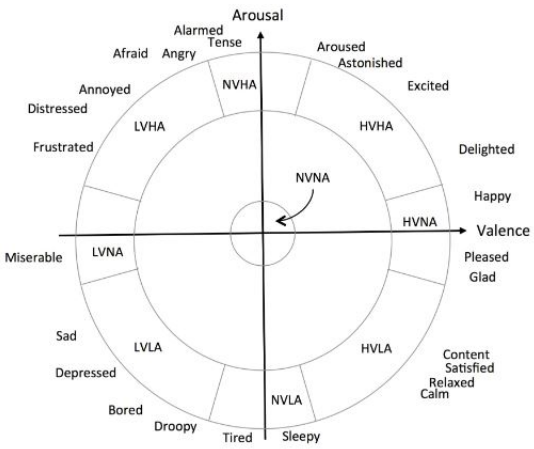


Figura 2: Escala de valencia-*arousal*.

de emoción tridimensionales que agregan el eje de "dominancia", termino que se refiere a que tanto control tiene sujeto sobre dicha emoción, esto facilita diferenciación de emociones como miedo y enojo.

Dada su simplicidad y baja complejidad de modelación, el modelo bidimensional de valencia-*arousal* es el preferido en el contexto de aprendizaje de máquina.

6.2. Psicología de la música

El estudio por parte de la ciencia cognitiva en el área de la música solía enfocarse en los orígenes de nuestro gusto por la música como especie,su aprendizaje y su perfección técnica y estética. [4]. Sin embargo, a lo largo de los años ha crecido el interés en un aspecto que es fundamental tanto a nuestra experiencia de escucha como el desarrollo de nuestras tendencias musicales, los sentimientos que esta genera [5]. En una encuesta realizada en 2015, la mayor razón que los hombres y mujeres reportaban para escuchar música era 'crear/mantener una emoción, sentimiento, expresión o estado anímico' [4].

El estudio de las emociones en la música busca entonces conocer la respuesta a dos preguntas principales.

- Cómo la música logra expresar emociones.
- Cómo la música logra hacer surgir emociones en los seres humanos que la escuchan.

Estas preguntas suelen ser respondidas en otras formas de arte también, pero presentan un problema especialmente grande en el campo de la música (la puramente instrumental), pues suele ser que las emociones van dirigidas hacia algo concreto, y la música es un medio mucho más abstracto.

6.3. Análisis de señales de audio

Existe un gran interés en el desarrollo y utilización de algoritmos de aprendizaje para análisis visual, de lenguaje y, recientemente, de señales de audio. Alexa, Siri, Google Assistant y aplicaciones móviles como Shazam son algunas de las aplicaciones comerciales del análisis de señales de audio que han generado más notoriedad en los últimos años, pero su utilidad va mucho más allá, y puede ser explotada en una multitud de industrias.

Las señales de audio son digitalizadas utilizando una técnica conocida como muestreo, el muestreo consiste en tomar la señal de audio y capturar la amplitud de la onda cada cierto tiempo. Este tiempo es un intervalo discreto que se conoce como tasa de muestreo, una tasa de muestreo alta genera menos pérdida de información, pero viene al costo de espacio y procesamiento. Al grado de precisión con el que se mide la amplitud de onda se le llama rango dinámico y está medida en bits. Por ejemplo, un rango dinámico de 16 bits cuenta con 65,536 valores posibles para la amplitud.

6.3.1. Pre-procesamiento

Lastimosamente la amplitud, frecuencia y la tasa de muestreo no ofrecen suficiente información para analizar una señal de audio. Por lo que es necesario derivar de ellas otras características que utilizar para modelarlas. Para empezar, las señales de audio pueden representarse dentro de un ambiente de análisis como líneas de tiempo a las que se les pueden realizar transformaciones.

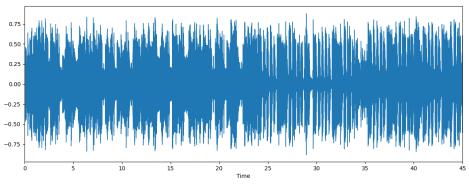
6.3.2. Espectrogramas

Los espectrogramas son maneras de representar la intensidad de la señal a lo largo del tiempo a través de varias frecuencias. Suelen presentarse los espectrogramas como mapas de calor, siendo los niveles de "energía" diferentes colores, como se puede observar en la Figura 3b.

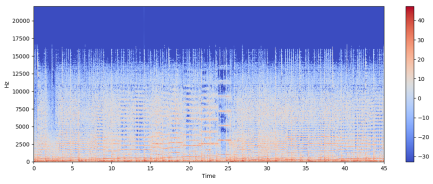
6.3.3. Spectral centroid

Es una característica que indica a qué frecuencia está centrada la energía del espectro, parecido a un 'centro de masa'. Se calcula como la media ponderada de las frecuencias en la señal, que han sido calculadas utilizando una transformada de Fourier con sus magnitudes como los pesos. La ecuación para calcularla es la siguiente:

$$f_c = \frac{\sum_k S(k) f(k)}{\sum_k S(k)}. \tag{1}$$



(a) Línea de tiempo para la canción id 1



(b) Espectrograma de canción id 1

En donde S(k) es la magnitud espectral de uno de los fragmentos de la línea de tiempo de frecuencias k y f(k) es la frecuencia central de ese fragmento.

6.3.4. Spectral Rolloff

Es la frecuencia bajo la cual reside un porcentaje predeterminado del total de la energía espectral.

6.3.5. Zero-Crossing Rate

Se utiliza para medir qué tan "suave" es una señal de audio, mide la cantidad de veces que una señal cruza el cero. Música con mayor cantidad de energía suele cruzar esta línea más veces, como el rock o el *heavy metal*.

6.4. Algoritmos supervisados

En aprendizaje supervisado, un modelo se crea con un conjunto de datos etiquetados que son utilizados para entrenar y verificar el desempeño del modelo. Existe una variable dependiente (el objetivo a predecir) y una o múltiples variables independientes de las cuales se pretende predecir el objetivo.

Los algoritmos de aprendizaje supervisados pueden separarse en clasificación y regresión.

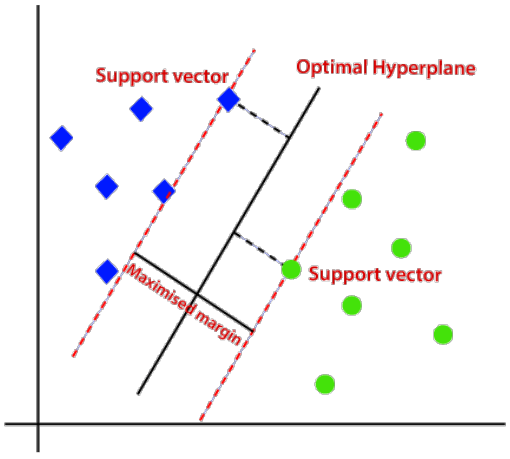
1. Clasificación: Asigna los datos de entrenamiento a una categoría discreta. Ejemplos de este tipo de algoritmos son clasificadores lineares, SVMs, y arboles de decisión.
2. Regresión: Se utiliza el algoritmo para entender las relaciones entre las variables dependientes e independientes,

estos algoritmos son efectivos para predecir valores continuos como precios. Algunos ejemplos de algoritmos de regresión son regresión linear y *SVR*.

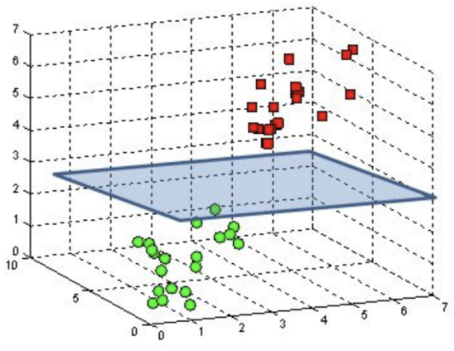
A continuación se describen algunos algoritmos supervisados que fueron considerados para la realización de el presente trabajo de graduación.

6.4.1. SVM

El método de *Support Vector Machine* asume un plano n-dimensional (n siendo el número de características) en donde ubica cada elemento del conjunto de datos. El objetivo del algoritmo es encontrar el hiperplano en este espacio n-dimensional que mejor separe el conjunto de datos en sus categorías. *SVM* soporta un margen de tolerancia como parámetro a su entrenamiento, un margen mas grande resulta en mejor clasificación de datos nuevos, pero puede causar problemas de convergencia en el entrenamiento; lo inverso es cierto para un margen muy pequeño, que tiende a crear sobre-ajuste [11]. La posición en el espacio del hiper-plano esta definida por los *support vectors*, puntos cercanos a la barrera de decisión de los cuales esta desea separarse, estos pueden visualizarse en la Figura 4a.



(a) *SVM* en un espacio 2D.



(b) *SVM* en un espacio 3D.

6.4.2. K-nearest neighbours

Este algoritmo hace uso de una función de similitud para calcular que tan parecidos son dos conjuntos de datos. Para clasificar una nueva entrada, el algoritmo la compara con todas las entradas ya existentes y determina a cuales se parece mas. Una vez encuentra cuales son las k entradas mas parecidas a la nueva (los *k-nearest neighbours*), clasifica la nueva entrada como la clase mas recurrente entre ellas [6]. El éxito de este algoritmo depende mayormente en dos aspectos: la eleccion de una funcion de similitud que sea adecue a los datos, y un valor apropiado de k. Una de las funciones de similitud mas recurrentes es la distancia euclidiana, cuya formula esta descrita en (2), en donde n se refiere al número de características presentes en una fila del conjunto de datos, x es el elemento con el que se esta comparando la nueva entrada y y es esta nueva entrada.

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2}$$

6.5. Redes neuronales

Las redes neuronales son algoritmos de aprendizaje creados en los años 80 y que, debido al creciente poder computacional, han empezado a ganar popularidad en los últimos 15 años. Una de sus ventajas es que, como las *SVM*, es capaz de modelar fronteras de decisión no lineales. La idea de una red neuronal es tomar las variables independientes y combinarlas con el fin de modelar una función no lineal que describa nuevas entradas [11].

Las redes neuronales están compuestas de capas de neuronas que toman inspiración del cerebro humano. La primera

capa es la capa de entrada, le siguen una o múltiples capas escondidas y por último está la capa de salida. Cada neurona se conecta a una o más neuronas de la siguiente capa y posee un peso asociado que determina la intensidad de la activación. Además del peso, las neuronas poseen un valor de activación mínimo; si este no es alcanzado, dentro de la misma no pasa información a la siguiente capa. Esto permite .ªpagar"neuronas [12]. Un ejemplo de cómo se ve una red neuronal se puede encontrar en la Figura 5

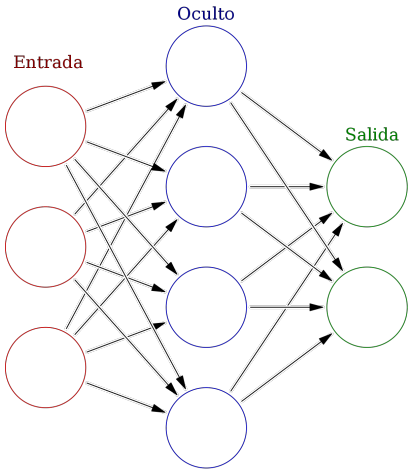


Figura 5: Red neuronal.

Dentro de las redes neuronales, existen también diferentes variaciones que se han desarrollado a lo largo de los años, algunas de estas están descritas más adelante.

6.5.1. Feedforward Neural Networks

Una de las formas más básicas de red neuronal, sus entradas son solo procesadas en una dirección hasta que llegan a la capa de salida. No existe *backpropagation*, son utilizadas por su relativa simplicidad y menor costo computacional que sus hermanas.

6.5.2. CNN

Parecidas a las redes neuronales *Feed Forward* descritas anteriormente, su información solo viaja en una dirección, pero dentro de sus capas existen algunas a las que se les llama çapas de filtro"(o *convolutional layers*), estas pueden estar completamente interconectadas o agrupadas. Antes de pasar el resultado a la siguiente capa, las capas de filtro aplican una operación de convolución a la entrada que reciben, gracias a esta operación las redes *CNN* necesitan menos parámetros y pueden ser mucho más *deep*.

Gracias a estas características, las *CNN* muestran especial utilidad en análisis de imágenes, vídeo, lenguaje natural y sistemas de recomendación.

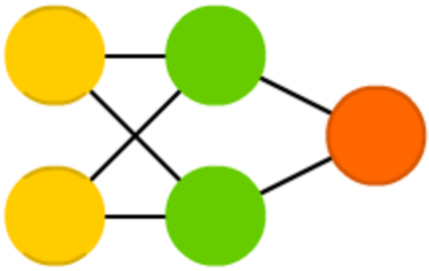
6.5.3. RNN

Este es un tipo de red neural en la que la salida de una neurona es guardada y posteriormente utilizada como entrada nuevamente. En una *RNN* cada paso toma en cuenta qué sucedió en el anterior, por lo que esta puede capturar la información secuencial presente en su entrada.

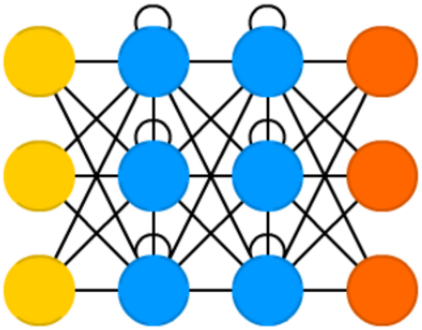
Las *RNN* comparten parámetros entre diferentes pasos de tiempo (iteraciones de entrenamiento), por lo que es posible utilizar menos parámetros y reducir el costo computacional. El tamaño del modelo no cambia con un input más grande, lo que nos permite procesar una entrada de cualquier tamaño.

Por otro lado, es común ver gradientes explosivos o que desaparecen con los pasos de tiempo, pues dada la naturaleza de las *RNN* los aumentos o decrementos del gradiente son multiplicativos que se salen de control, en especial si se tiene un número elevado de capas.

Una comparativa de comó se ve una *RNN* en comparación con una *Feed Forward NN* puede verse en 6b.



(a) *Feed Forward.*



(b) *RNN.*

CAPÍTULO 7

Marco metodológico

7.1. Obtención de datos

Uno de los retos más prominentes en la detección de emociones en piezas musicales es la falta de *datasets* ordenados y clasificados, por lo tanto, la primera tarea que se debió solventar fue la obtención de un *dataset* que permitiera el entrenamiento de los algoritmos de aprendizaje. Las dos opciones más prominentes en el campo son: *MoodyLyrics* que contiene alrededor de 2500 canciones clasificadas a través de *Amazon Mechanical Turk* en 4 categorías 'feliz', 'triste', 'enojado' y 'relajado' y *1000 songs Database* [13], que como su nombre lo indica es una recolección de canciones seleccionadas de *Free Music Archive (FMA)* y que fueron anotadas utilizando la misma plataforma que *MoodLyrics* (Es decir, mediante *Amazon Mechanical Turk*), pero utilizando la escala de valencia-*arousal*, las canciones fueron clasificadas de manera separada para valencia y *arousal*.

Se seleccionó *1000 songs Database* sobre *Moodlyrics* dado que la escala valencia-*arousal* permitía acercarse al problema

de diferentes maneras sin estar atados a una clasificación predeterminada. Además, este conjunto de datos está anotado de manera continua, lo que permitió separar las canciones en intervalos sin necesidad de recapturar los datos para cada una de las mismás. El conjunto de datos también pone a disposición de los investigadores archivos mp3 de todas las canciones, que, como se explica más adelante, fueron utilizados en la generación de espectrogramás y extracción de *features*.

Adicionalmente, *1000 songs Database* provee media y desviación estándar, tanto de valencia como *arousal* para cada una de las canciones.

7.2. Lenguaje

La siguiente decisión que se enfrentó fue el lenguaje en el que se pretendía procesar los datos e implementar los algoritmos de aprendizaje. Por simplicidad, se decidió que un solo lenguaje debía cumplir ambas funciones, para no duplicar el posible mantenimiento y curva de aprendizaje. Dos lenguajes cumplían con los requerimientos del proyecto dada su relativa simplicidad, amplia comunidad científica creando recursos de aprendizaje de máquina y multitud de librerías y paquetes para lidiar con la manipulación de datos. Estos lenguajes eran:

- **R** Un lenguaje y entorno de desarrollo enfocado en computación estadística. *R* es de código abierto y un proyecto de GNU basado en *S*, este lenguaje es utilizado ampliamente en aprendizaje de máquina, estadística, minería de datos y matemática financiera. Cuenta con módulos que permiten la manipulación de datos y múltiples librerías creadas por la comunidad.
- **Python** Es un lenguaje de código abierto que tiene múltiples usos, a pesar de no estar enfocado en la estadística y graficación, la comunidad a gravitado a este lenguaje dada su versatilidad y su baja barrera de entrada. *Python*

también cuenta con multitud de módulos y librerías de manipulación de datos y algoritmos de aprendizaje.

Para el presente proyecto se decidió utilizar *Python* debido a que es un lenguaje mucho más versátil que *R*, las implementaciones de los algoritmos de aprendizaje están altamente optimizadas y tienen el apoyo de empresas pioneras en el campo como *Google* y *Facebook*, posee un manejador de paquetes (*pip*) más intuitivo y rápido, es un lenguaje más robusto y elegante, y puede utilizarse para otras etapas del proyecto haciendo posible mantener un solo lenguaje para la completitud del mismo.

7.3. Transformación de datos

A pesar de que los datos presentes en *1000 songs Database* [13] son una buena base para la investigación, fue necesario realizar algunas modificaciones y adiciones con el fin de hacerlos utilizables para los algoritmos de aprendizaje. En esta sección se explican con mayor detalle estas modificaciones y su implementación, así como las herramientas que hicieron posible las mismás.

7.3.1. Librosa

Librosa [14] es una librería de python construida para el análisis de audio y música que provee multitud de funcionalidades, tanto de preprocesamiento como de extracción de *features* y filtrado. A lo largo del trabajo, esta librería fue utilizada para crear el dataset utilizado con base en los archivos mp3 que se obtuvieron de *1000 songs Database* [13].

7.3.2. Preprocesamiento

Antes de extraer *features* del conjunto de datos, las señales de audio deben pasar por tres procesos,

- **Muestreo.** con el fin de homogeneizar el audio de las canciones y eliminar un posible factor de error, la tasa de muestreo de todas las canciones se fijó en 44100Hz, esto se hizo al cargar las canciones para su modificación con *librosa*.
- **Normalización.** Se utilizó *librosa* para normalizar la señal de audio y así eliminar picos excesivos que pudieran afectar el rendimiento de los algoritmos de aprendizaje.
- **Creación de un espectrograma.** Esta representación de la canción puede ser utilizada para obtener *features*, para obtenerla se utilizó la función *specshow* de *librosa*.

7.3.3. Extracción de *features*

Para el entrenamiento de los algoritmos de aprendizaje, no basta con conocer su amplitud y su intensidad, es necesario extraer de estos arreglos información adicional. Las características espectrales se obtienen de utilizar la transformada de Fourier para convertir la señal de audio. *Librosa* presenta una vasta cantidad de funciones que nos permiten extraer esta información. Para esta investigación se extrajeron las siguientes *features*:

- ***Chroma shift***
- ***Spectral centroid***
- ***Spectral Bandwith***
- ***Spectral Rolloff***
- ***Zero Crossing Rate***

- ***MFCCs***: Dados los datos, se decidió limitar los *MFCCs* a 30, pues extraer más podría hacer que el modelo fuera más complejo de lo necesario.

7.3.4. Limpieza

Durante el análisis preliminar fueron detectadas canciones duplicadas dentro del *dataset*, pues estas producían exactamente las mismás *features*, espectrogramás y valores de valencia. Estos datos duplicados fueron eliminados, el número de canciones restantes fue de 744. Este fue el primer *dataset* con el que se entrenaron y probaron los algoritmos de aprendizaje descritos más adelante.

La librería de manejo de datos *pandas* cuenta con la función *drop duplicates*, esta facilitó la eliminación de los mismos.

7.3.5. Conversión de variable objetivo

Con la finalidad de utilizar los algoritmos de clasificación, se convirtieron los valores numéricos continuos de valencia y *arousal* a 4 categorías de emoción discretas: felicidad, relajación, enojo y aburrimiento.

CAPÍTULO 8

Experimentación

En este capítulo se describe la implantación de los experimentos de detección de emociones, sus generalidades y posteriores optimizaciones.

8.1. Implementación y entrenamiento de algoritmos de aprendizaje

Todos los modelos fueron probados con el mismo conjunto de datos para asegurar que sus resultados fueran comparables. Además se utilizó el mismo lenguaje de programación, *Python*, para evitar atribuir discrepancias de rendimiento a optimizaciones hechas por un lenguaje de programación diferente.

Para todos los algoritmos, el conjunto de datos fue separado en dos partes: 80 % para entrenamiento y 20 % para validar el desempeño del mismo con entradas que este no había visto antes.

8.1.1. Librerías y herramientas

Para los modelos de *SVM* y *SVR* se utilizó *Scikit-learn* [15] en su versión 0.24.2. *Scikit-learn* es una librería de código abierto escrita en *Python*, publicada en 2010, cuyo propósito es ser una abstracción de algoritmos de aprendizaje de máquina. Soporta múltiples algoritmos de regresión, clasificación, y clusterización.

Fue diseñado para operar sin problemás con *Numpy* y *Scipy*, librerías de procesamiento, operación y manejo de matrices.

En cuanto a la Red neural, la librería elegida fue *Keras* dada su facilidad de integración con las herramientas que ya estaban siendo utilizadas en el proyecto y su simplicidad de implementación.

Para el manejo del conjunto de datos se utilizó *Pandas* y todas las gráficas generadas en el proyecto fueron extraídas utilizando *Matplotlib*.

8.1.2. SVM

La *support vector machine* fue implementada utilizando *SVM* de *Scikit-learn*. Con la finalidad de encontrar el valor de c (el ancho de la tolerancia en el hiperplano), se iteró sobre un espacio lineal de posibles c's creado con la ayuda de la función linspace de *numpy*.

El algoritmo se entrenó por un máximo de 10,000 iteraciones.

8.1.3. SVR utilizando valencia y *arousal* en modelos separados

Dado que la variable objetivo original era un valor continuo, existía la posibilidad de utilizar modelos de regresión para

predecirlos si se utilizaba como entrada los valores originales de valencia y *arousal*.

Con la finalidad de descubrir que acercamiento era mejor, se hizo precisamente eso. Para empezar se plantearon dos modelos, uno entrenaría sobre valencia y el otro sobre *arousal*, de manera separada hasta que fueran unificados en el último paso. La variable objetivo de estos modelos es el valor numérico de valencia y *arousal* respectivamente.

Los pasos a seguir para el entrenamiento de estos modelos fueron similares a aquellos seguidos por el modelo *SVM*. Implementado utilizando *LinearSVR* de *Scikit-learn* y encontrando el valor óptimo de c de la misma manera que en el anterior, iterando sobre una espacio linear de posibles c's creado con la ayuda de la función linspace de *numpy*. Nuevamente entrenando el modelo por un máximo de 10,000 iteraciones.

Una vez entrenados estos dos modelos de *SVR*, sus predicciones continuas de valencia y *arousal* fueron transformadas en categorías discretas de emoción para ser comparados con los modelos de clasificación.

8.1.4. Red neural

La Red neural fue programada en *keras* con ayuda del modelo *Sequential*. La topología de Red neural que se utilizó fue una capa de input para todas las características del conjunto de datos, dos capas ocultas con 256 neuronas y función de activación *relu* y una capa de salida con 4 neuronas y función de activación *softmax*.

Para este modelo se utilizó la función de pérdida de entropía cruzada, la métrica de éxito fue establecida como precisión y se compilo con el optimizador *adam*. Además, se entrenó a lo largo de 500 épocas y con un tamaño de lote de 32.

8.2. Optimizaciones

Tras el entrenamiento de los algoritmos descritos anteriormente, se tomaron acciones con la finalidad de mejorar su desempeño. Estas acciones (excluyendo cambios de hyperparametros) son descritas en la presente sección.

8.2.1. Interfaz web de realimentación

La primera forma en la que se intentó mejorar los algoritmos fue mediante la obtención de mayor cantidad de datos. Para este propósito se diseño una aplicación web y componentes que permitieron la recolección de las respuestas que los participantes proveyeron. Todos los componentes fueron alojados en *AWS* en su metodología gratis. *AWS* permite a los usuarios nuevos 12 meses de beneficios limitados para sus servicios de manera gratuita. Para este proyecto el importe total fue de 11.50$, importe cobrado por la zona alojada para el dominio *rodrigoalva.net*, sobre el cual se registraron tanto la página web como el *API* y el costo del dominio mismo en el registro de *Amazon*.

El dominio .net fue adquirido con la finalidad de que fuera útil para todos los servicios necesarios para el proyecto y permitiera emitir certificados SSL para los mismos. Esto elimina mensajes de precaución de seguridad en los navegadores y aumentó la posibilidad de que un usuario sin conocimiento informático llenara el formulario sin preocupaciones.

Los componentes de la interfaz web son los siguientes:

- ***Webpage***: Programada en *Javascript* con la librería de desarrollo de *frontend React*. Cada vez que se carga la página web esta elije un número aleatorio entre 1 y 733 (el número de datos) y pide al *API* la información de la canción seleccionada, esta incluye su calificación de valencia y *arousal* así como su nombre y su locación en *Amazon*

S3. Posteriormente extrae un fragmento de la canción que esta subida en *Amazon S3* para que el usuario la escuche y presenta su clasificación. Una vez el usuario ha decidido si está de acuerdo con la calificación o no e ingresa nuevos valores para la clasificación, el sitio web envía estos valores al *API* para su recolección.

El alojamiento para la página web utiliza *Amazon Amplify*, un servicio que permite conectar *AWS* con una *pipeline* de integración de *Github* y desplegar el proyecto asociado con un dominio y certificado de *Route 53*. Esta tecnología permitió desplegar cambios al proyecto de manera rápida y sencilla, además de dejar libre el limite de instancias *EC2* gratis para el modulo de *API*.

- ***API***: Con la finalidad de mantener el proyecto lo más homogéneo posible en términos de lenguajes de programación, este módulo fue programado en *Python* con la ayuda de *FastAPI*, una librería que abstrae el manejo de peticiones y permite crear *APIs* idiomáticas en *Python* de manera rápida y simple. Esta librería no requiere sobreponerse a una curva de aprendizaje empinada una vez se es familiar con *Python* y la manera en la que este lenguaje es escrito.

  Para este proyecto se creó el modelo **Song**, una abstracción de la tabla de datos que se tiene en *RDS*, utilizando este modelo se crearon dos rutas, la primera recibe un id de canción y devuelve su calificación de valencia y *arousal* así como su nombre y su locación en *Amazon S3*, esta ruta la utiliza la página web para mostrar la información inicial al usuario. La segunda ruta recibe un id de canción, un valor para valencia y uno para *arousal*, esta ruta guarda estos valores como una nueva entrada en la base de datos con la finalidad de utilizarlos para el entrenamiento de los algoritmos.

- ***Almacenamiento RDS y S3***: *RDS* es un servicio de *Amazon* que provee instancias de bases de datos en la nube. Este proyecto utilizó PostgreSQL 12.7 como base de datos. Se utilizó *RDS* para mantener la información de las canciones, su calificación, nombre y una referencia a su locación en *S3*. *S3* es un servicio de almacenamiento basa-

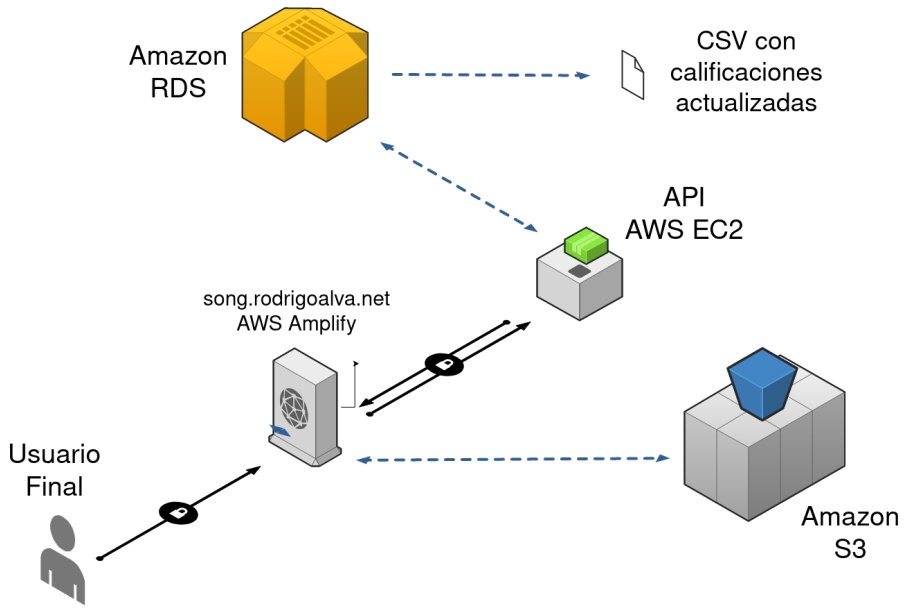


Figura 7: Flujo de funcionamiento de interfaz web.

do en *buckets* (contenedores fundamentales que funcionan como carpetas) en donde se almacenaron fragmentos de las canciones del conjunto de datos.

Este sistema permitió la adición de 45 datos al conjunto de datos original. Una visualización del flujo de funcionamiento de esta herramienta puede verse en la Figura 7

8.2.2. Eliminación de datos ambiguos

Dada la naturaleza de la escala valencia-*arousal*, era posible que existieran filas dentro del conjunto de datos cuya calificación afectara de manera negativa al entrenamiento de los algoritmos de aprendizaje, Si un elemento se encuentra en "el centro" de la escala valencia-*arousal*, se teorizo podría ser confuso para el modelo asignarlo a una categoría, pues no pertenece fuertemente a ninguna. Con el fin de probar si la presencia de estos datos hacia alguna diferencia en los resultados finales, todos los algoritmos fueron probados con el conjunto de datos original y el nuevo conjunto de datos eliminando los datos ambiguos y posteriormente comparados.

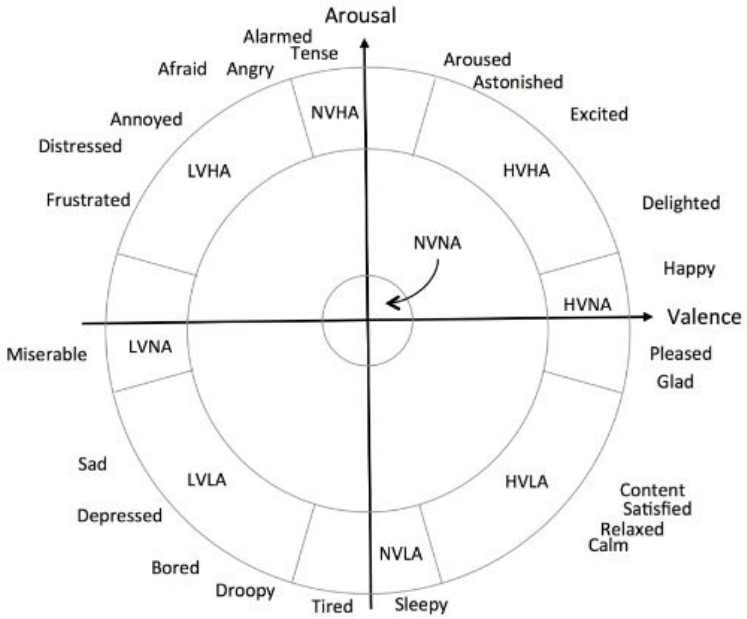


Figura 8: Escala valencia-*arousal* con inatención en el centro.

La escala valencia-*arousal* va desde -1 a 1 en ambos ejes. Para el conjunto de datos de prueba se eliminaron aquellos elementos que tuvieran una valencia y arousal entre -0.1 y 0.1. Esto eliminó un total de 230 filas del conjunto de datos original.

CAPÍTULO 9

Resultados y Discusión

9.1. Análisis de resultados

9.1.1. Problemás generales

A lo largo del proyecto, surgieron múltiples problemás que afectaron cada uno de los algoritmos de aprendizaje de diferente manera. En esta sección se analizaran aquellos que afectaron a la generalidad de los mismos, independientemente de cual de ellos se implementaron.

La primera y más importante limitación probo ser la cantidad de datos disponibles para entrenamiento y pruebas. A pesar de que se intentó mitigar este problema mediante la adición de datos a través de la encuesta descrita en la sección de metodología, los datos adicionales obtenidos no fueron suficientes para solventar de manera satisfactoria sus consecuencias adversas. Este problema esta fuertemente evidenciado en las matrices de confusión de la Figura 10, Figura 9 y en menor grado Figura 11, en las cuales se puede notar la falta de predicción para algunas clases. La realidad es que la cantidad

de canciones presentes en las categorías de emoción *bored* y *happy* excedía por un factor de al menos 2.5 a aquellas de *angry* y *relaxed*, por lo tanto los algoritmos aprendieron que para maximizar su precisión la mejor estrategia era obviar la predicción de estas categorías. Existen dos posibles acciones que podrían resultar en una mejora, la primera es generar nuevos datos ya sea mediante encuestas o mediante generación a partir del conjunto de datos original; la segunda es eliminar estas características pues solamente generan ruido para el resto del modelo.

9.1.2. Modelos

El modelo ***Support Vector Machine*** obtuvo un rendimiento de 62.35 % en el conjunto de entrenamiento y 59.30 % en el conjunto de prueba, como se ve en el Cuadro 1. En cuanto a la eliminación de datos considerados ambiguos, como se describió en la sección "Eliminación de datos ambiguos" del marco metodológico, este algoritmo obtuvo mejor rendimiento en el conjunto de entrenamiento con 70.35 %, pero un rendimiento considerablemente peor a su contra parte en el conjunto de pruebas con 54.30 %, evidenciado en el Cuadro 2. Esta discrepancia esta probablemente atribuida a *overfitting*. Es posible observar de la matriz de confusión para este algoritmo presentada en Figura 9, que este se desempeño especialmente bien en las etiquetas de emoción *bored* y *happy* además de ser el único algoritmo dentro de los 3 que predijo todas las clases al menos una vez.

A pesar de sus evidentes carencias, pues es simple ver que 62.35 % no es un rendimiento admirable, este algoritmo fue capaz de detectar el nivel de energía en la canción de manera correcta. Refiriéndose a la matriz de confusión en la Figura 9 se puede ver que sus fallos en las categorías de *angry* y *relaxed* (aquellas con menor cantidad de datos) fueron clasificados como su homónimo en el eje de valencia, pero manteniéndose en el mismo sector de *arousal*. 68 % de los datos de *relaxed* fueron clasificados en *bored* y 50 % de los datos que debían ser

angry se clasificaron como *happy*. Este resultado lleva a pensar que los indicadores para la intensidad de una canción son mucho más claros que aquellos relacionados a su connotación negativa/positiva.

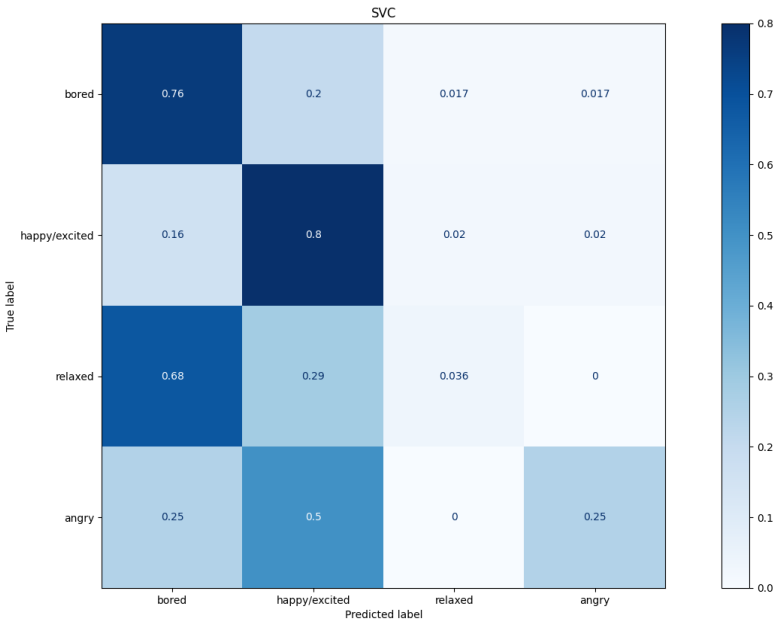


Figura 9: Matriz de confusión SVM.

SVM	**Accuracy**
Training	62.35 %
Test	59.30 %

Cuadro 1: Resultados *SVM* antes de eliminar datos.

SVM	**Accuracy**
Training	70.35 %
Test	54.30 %

Cuadro 2: Resultados *SVM* después de eliminar datos

La **Red neuronal** obtuvo un rendimiento de 61.85 % en el conjunto de entrenamiento y 55.70 % en el conjunto de prueba, como se ve en el Cuadro 3. En cuanto a la eliminación de datos considerados ambiguos, como se describió en la sección "Eliminación de datos ambiguos" del marco metodológico, este algoritmo obtuvo mejor rendimiento en el conjunto de entrenamiento con 66.85 %, pero un rendimiento considerablemente

peor a su contra parte en el conjunto de pruebas con 50.70 %, evidenciado en el Cuadro 4.

La Red neuronal fue el modelo que más sesgo sus predicciones a solamente dos clases, pues en el conjunto de pruebas *angry* y *relaxed* no fueron predichas ni una sola vez, tal y como se evidencia en su matriz de confusión en la Figura 10. De nuevo las tendencias generales de intensidad de emoción fueron predichas, sugiriendo que la Red neuronal fue capaz de aprender a detectar *arousal*. Dada la naturaleza de las redes neuronales, no es posible saber con certeza que se debe hacer para mejorar el rendimiento del algoritmo; a pesar de esto, existen intuiciones que se pueden aplicar. Algunas modificaciones que podrían resultar en un mejor desempeño son aumentar el ratio de *dropout* en las capas escondidas o simplificar la arquitectura (reducir el número de neuronas en cada capa).

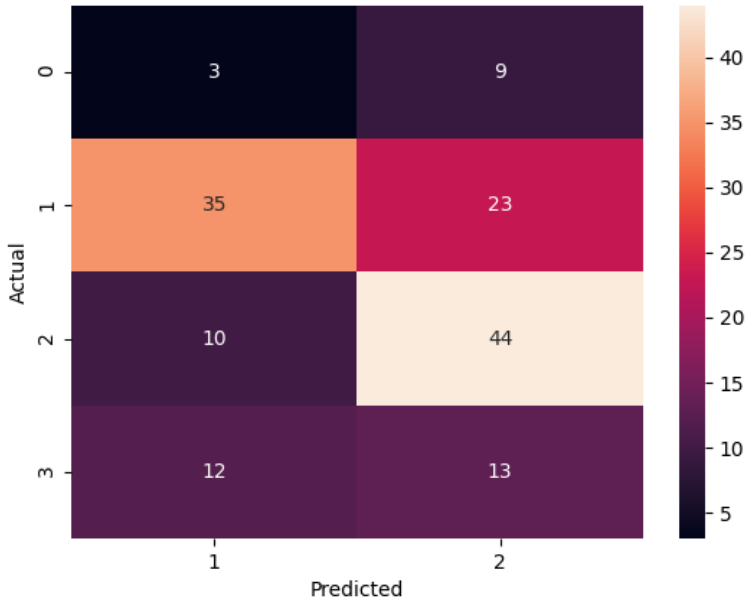


Figura 10: Matriz de confusión Red neural.

El último algoritmo a probar fue el ***Support Vector Regression***, este algoritmo fue implementado de manera diferente a los otros dos, pues es una combinación de dos algo-

Neural Network	**Accuracy**
Training	61.85 %
Test	55.70 %

Cuadro 3: Resultados *Neural Net* antes de eliminar datos.

Neural Network	**Accuracy**
Training	66.85 %
Test	50.70 %

Cuadro 4: Resultados *Neural Net* después de eliminar datos.

ritmos de regresión entrenados de manera separada, uno para valencia y uno para *arousal*. Estos valores iban desde -1 a 1 y se utilizó un ϵ de 0.1. El porcentaje de predicciones del conjunto de prueba dentro del parámetro ϵ para valencia fue de 88.59 % y para *arousal* de 66.44 %. Una vez combinadas estas predicciones se logró una precisión de predicción categórica de 50.03 %, evidenciado en el Cuadro 5. A pesar de no haber sido entrenado con datos categóricos, sino con los valores continuos de valencia y *arousal* este algoritmo nunca predijo la clase *angry*, de nuevo producto de la poca cantidad de datos. También produjo la precisión más baja entre los algoritmos probados. A pesar de estos factores, su porcentaje de predicciones dentro del valor ϵ en el análisis de valencia es un contraste interesante con la capacidad de los otros algoritmos de detectar *arousal*.

SVR	**% dentro de epsilon**	**Accuracy**
Valence	88.59 %	n/a
Arousal	66.44 %	n/a
Combinado categórico	n/a	50.03 %

Cuadro 5: Resultados *SVR* antes de eliminar datos.

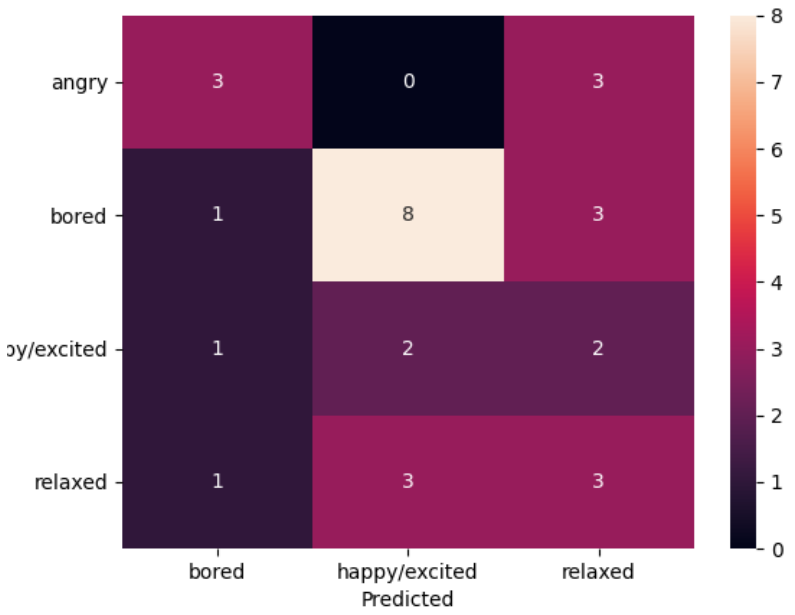


Figura 11: Matriz de confusión SVR.

SVR	**% dentro de epsilon**	**Accuracy**
Valence	80.59 %	n/a
Arousal	56.44 %	n/a
Combinado categórico	n/a	42.03 %

Cuadro 6: Resultados *SVR* después de eliminar datos.

En general, el mejor algoritmo implementado en este proyecto fue el ***Support Vector Machine***, pues este obtuvo el mejor porcentaje de aciertos además de ser el único modelo que produjo predicciones de todas las clases. Este resultado lleva a pensar que para este caso en particular, los algoritmos de clasificación supervisados podrían ser la mejor opción. Sin embargo, se considera prudente explorar con mayor detenimiento la implementación de dos modelos de regresión, pues con un mejor resultado de *arousal*, es posible obtener una precisión mucho mayor. Este acercamiento es especialmente flexible dada la capacidad de utilizar no solamente diferentes algoritmos para valencia y *arousal*, sino incluso conjuntos de datos con diferentes características extraídas.

La optimización eliminación de datos considerados ambiguos, como se describió en la sección “Eliminación de datos

ambiguos" del marco metodológico no mejoro la precisión de los modelos, de hecho causo *overfitting* y precisiones más bajas en todos. Lastimosamente esto es un efecto secundario de la baja cantidad de datos con la que se estaba trabajando para empezar. Es difícil concluir sobre la efectividad que hubiera tenido este método en un conjunto de datos más abundante.

CAPÍTULO 10

Conclusiones

- Fue posible generar un modelo de clasificación de emociones en piezas musicales con precisión mayor al 50 %. La red neuronal obtuvo una precisión del 55.7 % y la SVM del 59.3 %
- Para el conjunto de datos *1000 songs for emotional analysis* y las características extraídas, la detección de *arousal* es más efectiva en algoritmos de clasificación que la detección de valencia.
- El cuadrante correspondiente a felicidad es el predicho con mayor éxito.

CAPÍTULO 11

Recomendaciones

- Buscar la manera de generar mayor cantidad de datos para el conjunto de datos *1000 songs* con la finalidad de aumentar las posibilidades de un modelo con mayor *accuracy*
- Experimentar con más algoritmos de regresión separando valencia y *arousal*, pues a pesar de no ser el más preciso, el algoritmo de *SVR* fue el único que predijo las 4 clases en el conjunto de prueba. Es posible entonces que experimentación más profunda en esta forma de predicción logre producir mejores resultados. Se espera que este camino tenga el mayor margen de mejora si no es posible adicionar más datos al entrenamiento.
- Añadir *loudness*, *danceability* y *energy* a las características extraídas para cada canción, pues han demostrado ser de utilidad en trabajos como [6]. *Loudness* y *Energy* podrían ser significativos para la predicción de *arousal*. Mientras *danceability* presenta una característica interesante que es posible mejore la predicción de valencia. Estas características están disponibles en el API de Spotify.
- Analizar mayor número de MFCCs. El número de MFCCs extraídos en este proyecto fue elegido dado que su aumento no producía mejoras significativas al momento de entrenar los algoritmos de predicción. Aun así, es posible que diferentes arquitecturas de redes neurales o algoritmos no

probados en el presente trabajo se beneficien de las características extra.

- Expandir la gama de algoritmos. Solamente contando algoritmos de clasificación, es posible añadir al menos dos variantes que no han sido probadas en este proyecto y podrían presentar resultados interesantes, KNN y Random Forest.
- Implementar una LSTM. Esta arquitectura de red neural es comúnmente utilizada para reconocimiento del habla y composición musical, es posible que sus características sean de beneficio para la clasificación de emociones en el ámbito musical.

CAPÍTULO 12

Bibliografía

[1] K. Napier y L. Shamir, "Quantitative sentiment analysis of lyrics in popular music," *Journal of Popular Music Studies*, vol. 30, n.º 4, págs. 161-176, 2018.

[2] Y. Xia, L. Wang y K.-F. Wong, "Sentiment vector space model for lyric-based song sentiment classification," *International Journal of Computer Processing Of Languages*, vol. 21, n.º 04, págs. 309-330, 2008.

[3] J. Choi, J.-H. Song e Y. Kim, "An analysis of music lyrics by measuring the distance of emotion and sentiment," en *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, IEEE, 2018, págs. 176-181.

[4] S. de Boise, "Music Listening, Emotions, Age and Context," en *Men, Masculinity, Music and Emotions*, Springer, 2015, págs. 94-120.

[5] T. Cochrane, "Music, emotions and the influence of the cognitive sciences," *Philosophy Compass*, vol. 5, n.º 11, págs. 978-988, 2010.

[6] A. Jamdar, J. Abraham, K. Khanna y R. Dubey, "Emotion analysis of songs based on lyrical and audio features," *arXiv preprint arXiv:1506.05012*, 2015.

[7] M. Cabanac, "What is emotion?" *Behavioural Processes*, vol. 60, n.º 2, págs. 69-83, 2002, ISSN: 0376-6357. DOI: https://doi.org/10.1016/S0376-6357(02)00078-5. dirección: https://www.sciencedirect.com/science/article/pii/S0376635702000785.

[8] P. J. Bota, C. Wang, A. L. N. Fred y H. Plácido Da Silva, "A Review, Current Challenges, and Future Possibilities on Emotion Recognition Using Machine Learning and Physiological Signals," *IEEE Access*, vol. 7, págs. 140 990-141 020, 2019. DOI: 10.1109/ACCESS.2019.2944001.

[9] W. L. Rosensohn, "A Logical Method for Making a Classification of Emotions, Using Wilhelm Wundt's Theory of Emotion Formation," *The Journal of Psychology*, vol. 55, n.º 1, págs. 175-182, 1963. DOI: 10.1080/00223980.1963.9916610. eprint: https://doi.org/10.1080/00223980.1963.9916610. dirección: https://doi.org/10.1080/00223980.1963.9916610.

[10] P. Schmidt, A. Reiss, R. Duerichen y K. Van Laerhoven, "Wearable affect and stress recognition: A review," nov. de 2018.

[11] A. Akalin, *Computational Genomics with R*, ép. Chapman & Hall/CRC Computational Biology Series. CRC Press LLC, 2020, ISBN: 9781498781855. dirección: https://books.google.com.gt/books?id=KqW6swEACAAJ.

[12] I. C. Education, *Neural Networks*, ago. de 2020. dirección: https://www.ibm.com/cloud/learn/neural-networks.

[13] M. Soleymani, M. N. Caro, E. M. Schmidt, C.-Y. Sha e Y.-H. Yang, "1000 songs for emotional analysis of music," en *Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia*, 2013, págs. 1-6.

[14] B. McFee, V. Lostanlen, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, J. Mason, D. Ellis, E. Battenberg, S. Seyfarth, R. Yamamoto, K. Choi, viktorandreevichmorozov, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, D. Hereñú, F.-R. Stöter, P. Friesch, A. Weiss, M. Vollrath y T. Kim, *librosa/librosa: 0.8.0*, ver. 0.8.0, jul. de 2020. DOI: 10.5281/zenodo.3955228. dirección: https://doi.org/10.5281/zenodo.3955228.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot y E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, págs. 2825-2830, 2011.

CAPÍTULO 13

Anexos

13.1. Enlaces

El código fuente para los algoritmos de aprendizaje implementados en este proyecto puede encontrarse en: Github Otros enlaces relevantes pueden encontrarse en Cuadro 7:

Nombre	**Enlace**
Código fuente página web	https://github.com/alv16106/song-sentiment
Página web	https://song.rodrigoalva.net/

Cuadro 7: Enlaces.

13.2. Herramientas utilizadas

13.3. Imágenes página web

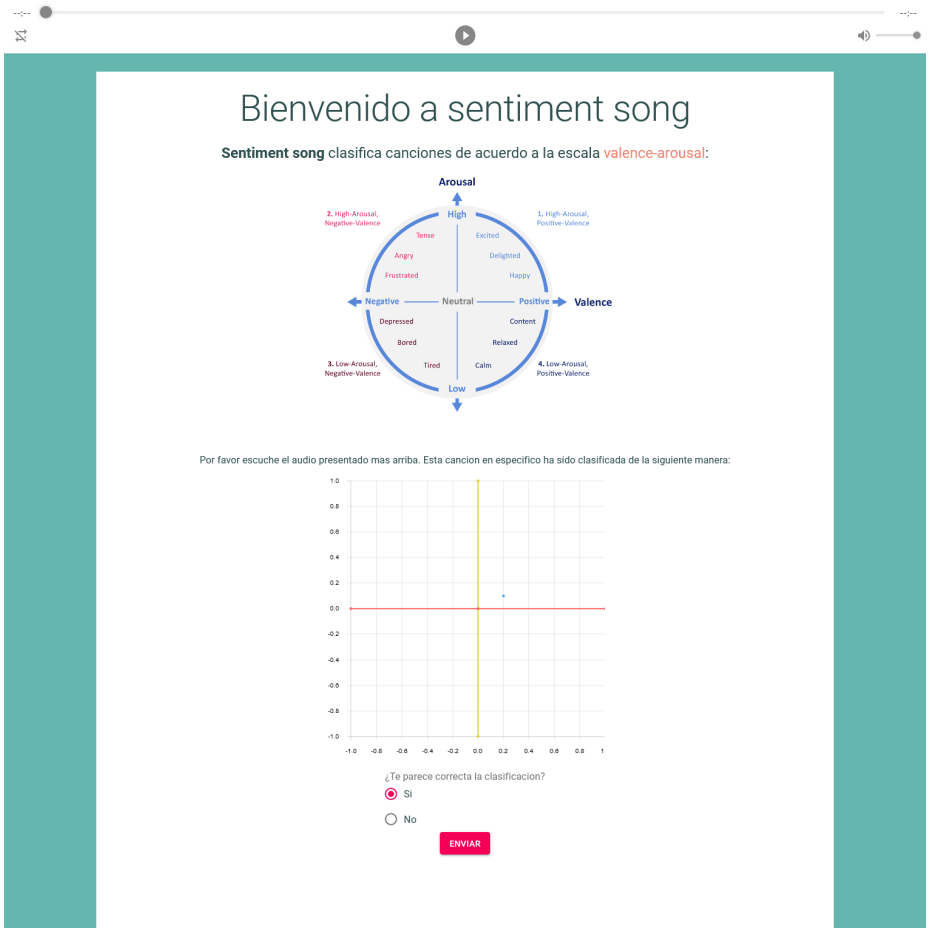


Figura 12: Página web cuando el usuario está de acuerdo con la clasificación.

Figura 13: Página web cuando el usuario no está de acuerdo con la clasificación.

CAPÍTULO 14

Glosario

arousal: Es el estado fisiológico y psicológico del despertar o de los órganos de los sentidos estimulados hasta un punto de percepción. 2