UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN INGENIERÍA MATEMÁTICA

APLICADA AL ANÁLISIS DE DATOS

PROYECTO FIN DE GRADO

EMoody: Detección de Emociones a través de la Voz

MARTA ALMENDRO ÁLVAREZ

Dirigido por

ANA DEL VALLE CORRALES PAREDES

CURSO 2020-2021

TÍTULO: AFFECTIVE COMPUTING: HERRAMIENTA DE ACOMPAÑAMIENTO BASADA EN EL ANÁLISIS DE EMOCIONES

AUTOR: MARTA ALMENDRO ÁLVAREZ

TITULACIÓN: GRADO EN INGENIERÍA MATÉMATICA APLICACA AL ANÁLISIS DE DATOS

DIRECTOR/ES DEL PROYECTO: ANA DEL VALLE CORRALES PAREDES

FECHA: JUNIO DE 2021

RESUMEN

Las emociones habitan en nuestro interior, como un fiel compañero que responde por nosotros a los estímulos externos. Se pueden definir de manera biológica como las reacciones psicofisiológicas que empleamos como modo de adaptación a lo que percibimos a nuestro alrededor. Desde una perspectiva un poco más filosófica, las emociones son la expresión de nuestros sentimientos, la afección del alma, o como dijo Descartes, las 'pasiones del alma'.

Este proyecto plantea una fusión entre lo antiguo y lo nuevo; integrando el estudio clásico de las emociones con las tecnologías emergentes utilizando de la Inteligencia Artificial (IA). Para ello se ha empleado el Deep Learning, una de las herramientas novel de la IA que simula la mecánica neuronal de la mente humana. A partir de este concepto se ha desarrollado un modelo basado en redes neuronales profundas artificiales que predice de manera eficaz las emociones de un usuario a través de ciertas características presentes en su voz.

La finalidad de este proyecto es ofrecer una herramienta capaz de realizar estas predicciones para diferentes usuarios. Para ello se han juntado todas las piezas del puzle en una única aplicación web con una interfaz de usuario capaz de dar respuesta a peticiones grabadas en tiempo real.

Palabras clave: Computación afectiva, aprendizaje profundo, clasificación de emociones a través de la voz, extracción de características, aplicación web, interfaz de usuario

ABSTRACT

Emotions lie within un, like a loyal companion that will look out on external stimuli for us. If defined in biological context, they are the psychophysiological reactions we use to adapt to what we perceive around us. From a philosophical point of view, emotions are the expression of our feelings, our soul's affection, or how Descartes once defined them, the 'passions of our soul'.

The following project sets out a fusion between the archaic and the new; how to meld classic studies about emotions with emerging technologies through Artificial Intelligence (AI). To gain this, Deep Learning, one of IA's novel tools that pursues mimicking the neurological mechanics behind the human mind, has been employed. Parting from this concept, an artificial neural network has been developed to efficiently predict a user's emotions from certain characteristics extracted from his/her voice.

The purpose of this project is to offer a tool capable of delivering these predictions to different users. All the puzzle's pieces are put together in one single web application infrastructure with capacity to serve real time recorded requests.

Keywords: Affective Computing, Deep Learning, Speech Emotion Recognition, Feature Extraction, web app, user interface

AGRADECIMIENTOS

A todos aquellos que me han apoyado, ayudado y animado durante esta etapa tan importante. Quiero agradecer a todos los profesores que han fomentado mi curiosidad y me han ayudado a construir los cimientos de mi futuro profesional. Agradezco, en especial, a mis padres por haber sido mi pilar estos años; por haberme transmitido esa ambición y la confianza en mí misma que la conlleva.

"Estamos hechos de polvo de estrellas. Somos una forma de que el universo se conozca a sí mismo"

Carl Sagan

TABLA RESUMEN

	DATOS
Nombre y apellidos:	Marta Almendro Álvarez
Título del proyecto:	Affective Computing: Herramienta de Acompañamiento Basada en el Análisis de Emociones
Directores del proyecto:	Ana del Valle Corrales Paredes
El proyecto ha implementado un producto: **(esta entrada se puede marcar junto a la siguiente)**	SI
El proyecto ha consistido en el desarrollo de una investigación o innovación: **(esta entrada se puede marcar junto a la anterior)**	SI
Objetivo general del proyecto:	Desarrollar una aplicación multi plataforma integrada con un sistema de reconocimiento de las emociones del usuario a través de su voz.

Índice

RESUMEN 3
ABSTRACT 4
TABLA RESUMEN 7
Capítulo 1. RESUMEN DEL PROYECTO 14
1.1 Contexto y justificación 14
1.2 Planteamiento del problema 14
1.3 Objetivos del proyecto 14
1.4 Resultados obtenidos 15
1.5 Estructura de la memoria 15
Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE 16
2.1 Estado del arte 16
2.2 Contexto y justificación 30
2.3 Planteamiento del problema 31
Capítulo 3. OBJETIVOS 34
3.1 Objetivos generales 34
3.2 Objetivos específicos 34
3.3 Beneficios del proyecto 35
DESARROLLO DEL PROYECTO 36
3.4 Planificación del proyecto 36
3.5 Descripción de la solución, metodologías y herramientas empleadas 41
3.6 Recursos requeridos 63
3.7 Presupuesto 63
3.8 Viabilidad 64
3.9 Resultados del proyecto 64
Capítulo 4. DISCUSIÓN 85
Capítulo 5. CONCLUSIONES 86
5.1 Conclusiones del trabajo 86
5.2 Conclusiones personales 86

Capítulo 6. FUTURAS LÍNEAS DE TRABAJO 87
Capítulo 7. REFERENCIAS 88
Capítulo 8. ANEXOS 94

Índice de Figuras

Figura 1 Sistema HMI: usuario humano, interfaz y máquina (Papetti,2013) 16

Figura 2 Concentración de mercado del sector del Affective Computing (Market, 2021) 17

Figura 3 Estrella de emociones de Plutchik (Donaldson, 2017) 19

Figura 4 Sistema SER tradicional (Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H., & Alhussain, T. , 2019) 20

Figura 5 Características Segmentales y Suprasegmentales (Anagnostopoulos, Iliou & Giannoukos, 2012) 22

Figura 6 Esquema de Machine Learning con extracción de características (Chuan-En Lin, 2020) 23

Figura 7 Principales métodos de selección de características en Machine Learning (Karaarslan, 2019) 24

Figura 8 Método tradicional de Machine Learning vs. Deep Learning (Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H., & Alhussain, T. , 2019) 25

Figura 9 Arquitectura DNN (Glory, H.A., Vigneswaran, C., Jagtap, S.S. et al., 2021)..... 26

Figura 10 Celdas y estructura de LSTM (Mittidal, 2019) 27

Figura 11 Página de entrada al programa Empath..... 28

Figura 12 Página principal de la aplicación web Interview Simulator..... 29

Figura 13 Plantilla de la UI de la app Vmote 29

Figura 14 Evolución del sentimiento del miedo en España durante el periodo del 1 del marzo 2020 hasta el 30 de abril 2020 (de Las Heras-Pedrosa C, Sánchez-Núñez P, Peláez JI., 2020) ... 31

Figura 15 Noticias relacionadas con el miedo y el Covid-19 en los medios de comunicación (de Las Heras-Pedrosa C, Sánchez-Núñez P, Peláez JI., 2020)..... 32

Figura 16 Pipeline principal detrás del sistema de reconocimiento de voz..... 41

Figura 17 Representación del funcionamiento detrás de la técnica de oversampling SMOTE (Indresh Bhattacharyya, 2018)..... 46

Figura 18 Proceso de extracción de los coeficientes MFCC durante el Feature Extraction (Gong, S., Dai, Y., Ji, J., Wang, J., & Sun, H., 2015)..... 48

Figura 19 Mel filter banks basis functions using 20 Mel-filters in the filter bank (Yusnita, M. A., Paulraj, M. P., Yaacob, S., Yusuf, R., & Shahriman, A. B., 2013)..... 48

Figura 20 Formula de la Transformada de Coseno Discreta (DCT) 49

Figura 21 Secuencia del preprocesamiento de datos 49

Figura 22 Arquitectura de un modelo híbrido 1D CNN + LSTM (Hamad, R. A., Yang, L., Woo, W. L., & Wei, B., 2020) 52

Figura 23 Arquitectura de la estructura seleccionada y las dimensiones de los datos de salida de cada capa 56

Figura 24 Esquema del script desarrollado para el modelado del algoritmo de reconocimiento de emociones 56

Figura 25 Estructura y flujo de datos de la aplicación web 58

Figura 26 Jerarquía del directorio donde se despliega la aplicación 59

Figura 27 Logo EMoody 59

Figura 28 Logos de las tecnologías empleadas 60

Figura 29 Esquema del script app.py (ejecución de la web app) 61

Figura 30 Arquitectura que sostiene a la interfaz de usuario 62

Figura 31 Diagrama de barras mostrado el conteo de cada emoción antes de aplicar oversampling a los datasets femenino (izquierda) y masculino (derecha) 65

Figura 32 Diagrama de barras mostrado el conteo de cada emoción después de aplicar oversampling a los datasets femenino (izquierda) y masculino (derecha) 65

Figura 33 Gráfico de onda de una mujer expresando la emoción miedo 66

Figura 34 Espectrograma para la emoción de miedo (mujer) 66

Figura 35 Gráfico de onda de un hombre expresando la emoción miedo 66

Figura 36 Espectrograma para la emoción de miedo (hombre) 67

Figura 37 Gráfico de onda de una mujer expresando la emoción felicidad 67

Figura 38 Espectrograma para la emoción de felicidad (mujer) 67

Figura 39 Gráfico de onda de un hombre expresando la emoción felicidad 68

Figura 40 Espectrograma para la emoción de felicidad (hombre) 68

Figura 41 Gráfico de onda normal para la emoción neutral 68

Figura 42 Gráfico de onda de la emoción neutral con ruido 69

Figura 43 Gráfico de onda de la emoción neutral con onda desplazada 69

Figura 44 Gráfico de onda de la emoción neutral con onda alargada 69

Figura 45 Gráfico de onda de la emoción neutral con velocidad reducida 70

Figura 46 Gráfico de onda de la emoción neutral con velocidad aumentada 70

Figura 47 Gráfico de onda de la emoción neutral con cambio de tono 70

Figura 50 Gráfico de líneas con los coeficientes MFCC para hombres y mujeres a lo largo del mismo periodo de tiempo 72

Figura 51 PCA: Sin separación por sexos, extracción de ZCR, Chroma Shift, MFCC, RMSV, Mel Spectogram 72

Figura 52 PCA: Separación por sexo, 50 MFCCs 73

Figura 53 Gráficas de Accuracy y Loss durante el entrenamiento del segundo modelo de la tabla 11 75

Figura 54 Gráficas de Accuracy y Loss durante el entrenamiento del cuarto modelo de la tabla 11 76

Figura 55 Comportamiento del Loss en función del Learning Rate (Rosebrock, 2021) 77

Figura 56 Curvas de Accuracy y Loss para el entrenamiento y validación del modelo final. Género mixto 78

Figura 57 Curvas de Accuracy y Loss para el entrenamiento y validación del modelo final. Género femenino 78

Figura 58 Curvas de Accuracy y Loss para el entrenamiento y validación del modelo final. Género masculino 79

Figura 59 Matriz de confusión para las predicciones de emociones del dataset femenino 79

Figura 60 Matriz de confusión para las predicciones de emociones del dataset masculino 80

Figura 61 Matriz de confusión para las predicciones de emociones del dataset mixto 80

Figura 62 Sistema de reconocimiento de emociones 81

Figura 63 Pantalla principal de la aplicación web EMoody 82

Figura 64 Pantalla principal después de haber grabado un audio y mostrando el dropdown para seleccionar género 82

Figura 65 Pantalla de predicción- enfado 83

Figura 66 Pantalla de predicción- descontento/ repugnado 83

Figura 67 Pantalla de predicción- feliz 84

Figura 68 Pantalla de predicción- miedo 84

Índice de Tablas

Tabla 1 Características temporales de una señal de audio 21

Tabla 2 Características espectrales de una señal de audio 21

Tabla 3 Descripción de la identificación de archivos de audio de la base de datos RAVDESS 42

Tabla 4 Descripción de la identificación de archivos de audio de la base de datos CREMA-D ... 44

Tabla 5 Técnicas de Data Augmentation, descripción y librería de Python utilizada para su aplicación 47

Tabla 6 Dimensiones de los diferentes conjuntos de datos utilizados 50

Tabla 7 Hiperparámetros del modelo final 1D CNN + LSTM 53

Tabla 8 Arquitectura seleccionada para la red neuronal artificial detrás del sistema final..... 54

Tabla 9 Listado y descripción de las librerías empleadas en Python durante el proyecto 57

Tabla 10 Estimación del presupuesto del trabajo realizado 64

Tabla 11 Arquitectura de los modelos entrenados para el conjunto de datos de género mixto y los resultados de validation accuracy y los para cada etapa 74

Tabla 12 Arquitectura de los dos últimos modelos entrenados para el conjunto de datos de género femenino y los resultados de validation accuray y los para cada etapa..... 74

Tabla 13 Arquitectura de los modelos entrenados para el conjunto de datos de género masculino y los resultados de Validation Accuracy y Loss para cada etapa 75

Capítulo 1. RESUMEN DEL PROYECTO

1.1 Contexto y justificación

El estudio de la relación entre las emociones y la salud en los humanos se remonta a la Antigüedad. En la antigua Grecia esta idea se sustentaba en el hecho de que el cuerpo y la mente debían estar en armonía. La filosofía “mens sana in corpore sano”, que se traduce como “mente sana en un cuerpo sano”, era uno de los pilares de la medicina hipocrática.

La comunicación hombre-máquina se ha vuelto cada vez más ‘habladora’: Alexa, Cortana, Siri [1], entre muchos otros sistemas inteligentes de diálogo, han triunfado en el mercado de consumo de manera destacada, pero ¿y si éstos realmente pudiesen detectar nuestras emociones y reaccionar a ellas como un humano? Esto haría de la comunicación hombre-máquina (HMI) una comunicación mucho más natural, efectiva y agradable. La disciplina de reconocimiento automático de las emociones humanas y estados afectivos a través de la voz es conocida como Speech Emotion Recognition (SER).

1.2 Planteamiento del problema

Debido a la actual crisis sanitaria del Covid-19, el contacto físico hoy en día supone un peligro grave para muchas personas, para las cuales el hogar es su día a día. Este día a día incluye el cuidado de la salud, de modo que, este proyecto quiere ayudar a crear una alternativa al contacto físico para proporcionar seguridad en momentos de mayor malestar emocional.

Este proyecto plantea una alternativa para mantener un estado físico y mental saludables desde la comodidad del hogar. Se dirige hacia un amplio abanico de potenciales usuarios, desde personas mayores a las que desean cuidar sus familiares a distancia, hasta trabajadores en empresas, estudiantes o para formar parte de plataformas de atención al cliente, explotando, sobre todo, el auge de la prestación de servicios en remoto.

1.3 Objetivos del proyecto

Este proyecto propone una aplicación de reconocimiento de emociones a través de la voz, siguiendo la dinámica de la interacción hombre-maquina y basándose en el campo del Affective Computing. Para ello se plantea desarrollar un algoritmo de reconocimiento, como estructura de red neuronal artificial, empleando diferentes estructuras y experimentando con diferentes parámetros. Una vez obtenido el modelo adecuado, se persigue construir una aplicación web que permita utilizar este modelo a través de una interfaz de usuario (UI) adecuada y un diseño que haga su uso lo más dinámico y *user-friendly* posible. Por último, conectar ambos componentes back-end y front-end en una arquitectura que pueda dar respuesta a peticiones en tiempo real.

1.4 Resultados obtenidos

Se consigue desarrollar un sistema de detección de emociones a través de una grabación de voz. Siendo una red neuronal de aprendizaje el núcleo de este sistema se ha construido una arquitectura automatizada de captación de voz, extracción de características, preparado de datos para el entrenamiento y predicción de emociones. Este sistema es capaz de devolver la emoción detectada en cuestión de segundos. Para poder servir los modelos finales a un público potencial en forma de aplicación práctica, se desarrolla un servidor Flask simple que acepta una solicitud POST y realiza un preprocesamiento del audio recibido y devuelve una predicción en formato de texto y con iconos visuales. Por último, se ha diseñado para el proyecto una imagen de Marca propia.

1.5 Estructura de la memoria

La estructura de la memoria corresponde a los siguientes apartados:

Capítulo 1: contiene un breve resumen del contexto del proyecto, el planteamiento del problema, los objetivos y los resultados.

Capítulo 2: pone en contexto el proyecto y se especifican aquellos aspectos que respaldan las técnicas estudiadas y aplicadas a través del estado del arte.

Capítulo 3: incluye una descripción detallada de los objetivos, general y específicos, y los beneficios del proyecto.

Capítulo 4: expone el desarrollo completo del proyecto, desde la planificación, las herramientas y recursos empleados, en este caso mayoritariamente computacionales, hasta la viabilidad, el presupuesto estimado y los resultados obtenidos, además del componente de innovación.

Capítulo 5: plantea una discusión de los resultados obtenidos.

Capítulo 6: incluye las conclusiones obtenidas a raíz de los resultados.

Capítulo 7: propone las futuras líneas de trabajo consideradas tras la finalización del proyecto.

Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE

2.1 Estado del arte

Human-Machine Interaction

Para poder emplear la tecnología en el campo de reconocimiento de emociones humanas, debe existir la comunicación entre un humano y una máquina. De aquí nace el concepto Interacción Hombre-Máquina (HMI), descrito como la interacción y comunicación entre un usuario humano y una máquina (sistema técnico dinámico), a través de una interfaz hombre-máquina [2]. Para esta última parte, hay que tener en cuenta tanto la User Interface (UI) y la User Experience (UX).

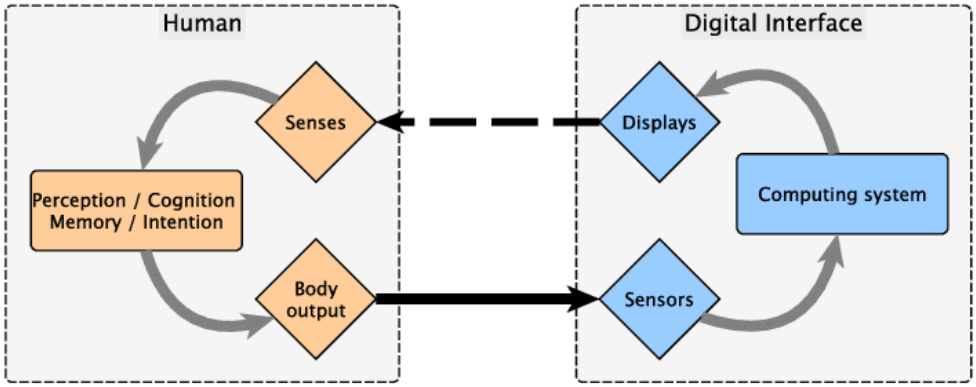


Figura 1 Sistema HMI: usuario humano, interfaz y máquina (Papetti,2013)

Affective Computing

Una de las grandes fuentes de inspiración de este proyecto viene de la idea del Affective Computing (AC). Este término fue propuesto por Rosalind Picard en 1997 como la computación que se relaciona, emerge o influye en las emociones [3]. Esto nos permite comunicarnos e interactuar con ordenadores, robots u otras tecnologías a través de nuestras emociones [4]. El AC persigue formar una base sólida en la investigación sobre la relación entre los estados afectivos, cognitivos y físicos del ser humano [5]. Un valor añadido de esta modalidad de computación es que se pueden acumular grandes cantidades de datos para el desarrollo de las ciencias cognitivas que servirán para el entrenamiento de tecnologías avanzadas que puedan, entre otras cosas, reforzar la capacidad de toma de decisiones. A través de tecnologías que monitorean y analizan las emociones, se pueden detectar amenazas a la salud tanto mental como física, incluso lograr reducir el problema de la detección tardía de enfermedades.

Según un informe de AllTheReasearch, los gigantes de la industria del AC son Google e IBM. Watson de IBM es uno de los pioneros en computación cognitiva orientada a la salud, con una de sus mayores aplicaciones en el campo de la oncología. Los productos se segregan entre

hardware y software y entre sus más destacadas aplicaciones está la sanidad. El segmento de la atención médica tiene algunas de las aplicaciones más avanzadas y comercializadas de AC. Empresas como DeepMind y Babylon Health están realizando importantes aportaciones al desarrollo de este campo [1]. La empresa Babylon lanzó en 2016 una aplicación de consultas médicas online; a través de un sistema de reconocimiento de voz y de inspección de historiales clínicos, ofrece una ruta de acción adecuada a la circunstancia médica en cuestión [6]. La contribución de Google en el sector es Google Health, una plataforma que promueve el descubrimiento de nuevas oportunidades en el campo de la AI con el fin de mejorar la eficacia de las tecnologías sanitarias a nivel global [7].

En cuanto al impacto que ha tenido la crisis del Covid-19 en este mercado, el uso generalizado de los dispositivos móviles y la penetración de internet en tantos rincones del mundo ha fomentado la inclinación progresiva hacia el uso de tecnologías digitales como el reconocimiento facial y de voz para mantener las conexiones de manera virtual. Además, aparecen las soluciones informáticas que permiten emplear estas tecnologías de reconocimiento para la detección de temperatura en el control de la propagación del virus. Muchas industrias están invirtiendo fuertemente en I+D para desarrollar software que ayude a controlar la propagación del Covid-19 [8].

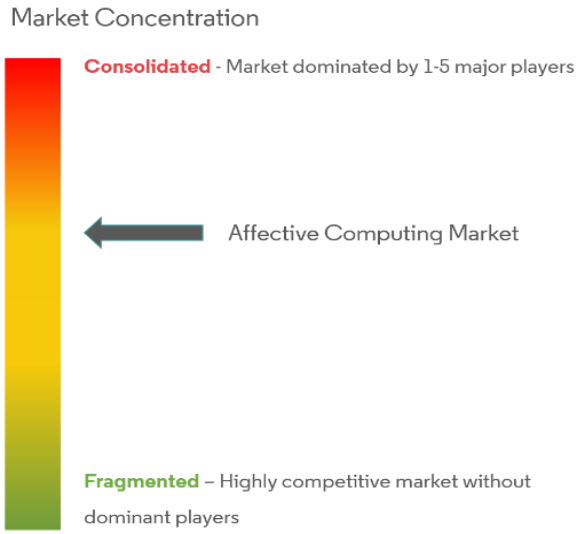


Figura 2 Concentración de mercado del sector del Affective Computing (Market, 2021)

Speech Emotion Recognition (SER)

Emplear las tecnologías en el reconocimiento automático de las emociones humanas y estados afectivos a través de la voz, normalmente conocido como Speech Emotion Recognition o SER, lleva siendo objeto de investigación varios años ya [9]. Ejemplo de ello son experimentos realizados (France et al. 2000) que identificaron propiedades acústicas del habla relacionadas con la depresión y riesgo de suicidio [10]. Otros campos donde se han empleado sistemas SER

son la educación, el entretenimiento, la industria automotriz y sistemas de síntesis de voz natural [11].

Los humanos poseemos una habilidad natural de emplear todos nuestros sentidos para obtener la máxima percepción de los estímulos y mensajes que recibimos. La detección de emociones, o empatía, es un proceso natural e innato en los humanos, pero es una tarea compleja para las máquinas ya que carecen de estas cualidades humanas. Por lo tanto, si el problema se desglosa, la tarea inicial es definir cómo se miden las emociones humanas. El estado emocional no puede ser reconocido de manera directa, por lo tanto, se queda en manos de las expresiones controladas por el sistema motor a través de varias formas; las principales son la voz y las expresiones faciales [12]. Las señales de voz son una de las formas más rápidas y naturales de comunicación humana [9]. Por ello, se convierten en el candidato ideal para servir de input en una interacción hombre-máquina más veloz y eficiente.

El primero paso requiere de un modelo apropiado de representación de la emoción través de clases discretas, en el que las emociones son categorizadas. Por lo general, se admite un grupo básico e innato de categorías emocionales, o patrones de respuesta emocional, esenciales en la supervivencia y evolución de la especie. Darwin (1872/1998) sugirió que las emociones se han desarrollado para servir una función de comunicación en la especie, y por tanto se expresan y reconocen de manera parecida de manera intercultural [13].

El grupo de emociones primordiales son conocidas como las "seis grandes" de Ekman (1992): sorpresa, alegría, ira, miedo, asco y tristeza [13]. A su vez, tenemos el concepto de ‘tonos emocionales’, desarrollado por Manfred Clynes, que se refiere a los siete tonos/energías emocionales que transmitimos los seres humanos a través de una sutil modulación del sistema motor [14]. Plutchik (1980) sugirió ocho emociones (enfado, anticipación, felicidad, confianza, miedo, sorpresa y tristeza) a través de un modelo tridimensional que modela los niveles de intensidad y las relaciones entre las emociones [15].

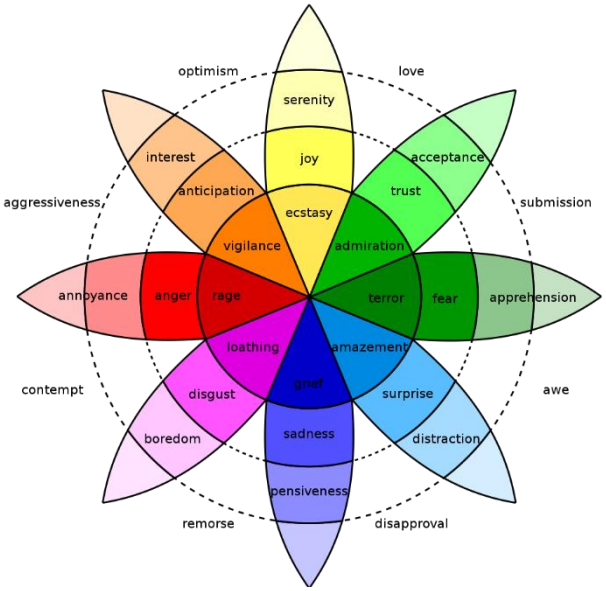


Figura 3 Estrella de emociones de Plutchik (Donaldson, 2017)

Uno de los mayores problemas a los que se enfrenta la disciplina del SER es que las emociones son conductas complejas, y cada persona siente emociones diferentes, y por lo tanto tiene formas distintas de mostrarlas [16]. Las emociones se manifiestan y pueden medirse atendiendo a tres niveles o componentes diferentes [17]:

- **Cognitivo**: por ejemplo, a través de las expresiones verbales que manifiestan los sentimientos o experiencia subjetiva.
- **Conductual o motor**: por medio de cambios faciales o conductas de aproximación o retirada.
- **Fisiológico**: a través de los cambios viscerales u hormonales, y de los de tipo eléctrico y metabólico que se dan a nivel cerebral y periférico.

El enfoque del SER está compuesto principalmente por las fases conocidas como la extracción de características (*Feature Extraction*), la selección de estas características (*Feature Selection*) y la clasificación de las características (*Emotion Classification*) [18].

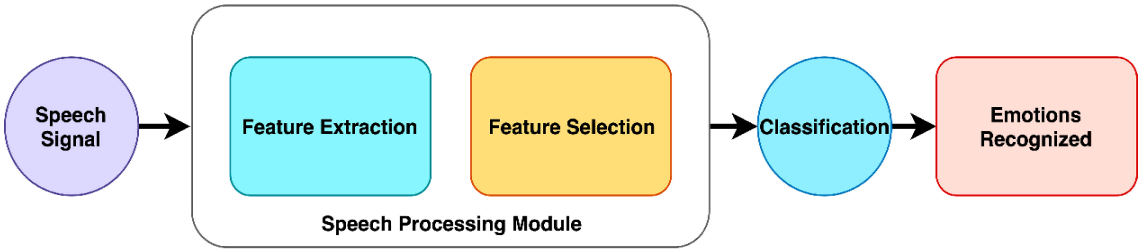


Figura 4 Sistema SER tradicional (Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H., & Alhussain, T. , 2019)

SER: Feature Extraction

El caso del tono de voz se cataloga como una respuesta fisiológica [6]. Realizar un reconocimiento de las emociones a través de la voz implica encontrar la manera de identificar estas emociones dentro de las señales de voz. Sin embargo, estas señales son complejas y almacenan información variada; el mensaje, quien habla, el género, el idioma… El SER es una tarea complicada ya que los tonos de voz son únicos para cada individuo [19]. Esta alta variabilidad del tono y la frecuencia puede resultar en que una misma frecuencia en cierto punto sugiera dos o más emociones. Por lo tanto, diferenciar entre las diferentes porciones dentro de una misma onda es una tarea complicada.

Un modelo no puede entender la información recibida en formato de audio, por lo tanto, es necesario convertir estos datos en un formato entendible. Para ello se utiliza la extracción de características, conocido como *Feature Extraction*. El *Feature Extraction* pretende extraer características particulares para cada una de estas porciones, denominadas *speech utterances* [20].

Las características del audio se categorizan generalmente en:

Características temporales

Estas características son fáciles de extraer, ya que se extraen directamente y por lo tanto requieren de menor poder de cómputo. Estas características proporcionan una manera más sencilla de analizar las señales de audio [21]. Estas incluyen:

Característica	**Descripción**
Zero-crossing rate	Tasa a la que una señal cambia de positivo a cero a negativo o de negativo a cero a positivo
Short-term energy	Calcular la cantidad de energía en un sonido en un momento específico (para distinguir el habla del silencio)

Maximum amplitude	Amplitud máxima del movimiento oscilatorio
Minimum energy	Energía mínima detectada
Entropy energy	Medida de la dispersión de la energía

Tabla 1 Características temporales de una señal de audio

Características espectrales

Son las relacionadas con el tracto vocal y suelen representar la distribución de la energía en la frecuencia del habla. Revelan patrones más profundos de las señales de audio, por lo que son las características que resultan más determinantes para determinar las emociones subyacentes [21].

Característica	**Descripción**
MFCC	Coeficientes Cepstrales en la Escala de Mel- representan el habla en torno a la percepción auditiva del ser humano.
LPCC	Coeficientes Cepstrales de Predicción Lineal- modelo basado en una imitación matemática del tracto vocal.
Formants	Picos de intensidad en el espectro de un sonido
DFT	Calcula el espectro de frecuencia de una señal. Esto permite que los sistemas calculen en el dominio de la frecuencia.
Spectral centroid	Indica dónde se encuentra el centro de masa del espectro.
Linear prediction	Método utilizado para representar el envolvente espectral de una señal de forma comprimida utilizando un modelo predictivo lineal.
Chroma features	Herramienta especial para analizar música, categorizando los tonos en 12 tonos diferentes.

Tabla 2 Características espectrales de una señal de audio

Características prosódicas

En muchos idiomas, son las responsables de transmitir información semántica al oyente, y forman las bases del comportamiento lingüístico. Por lo tanto, son útiles para detectar las emociones del locutor [22]. En varias investigaciones, estas características se han empleado para tareas de reconocimiento de emociones [23] [24].

- **noise**
- **stretch**
- **shift**
- **pitch**
- **higher_speed**
- **lower_speed**

Estas se explican en detalle más adelante a la hora de emplearlas en el modelo.

Los vectores de características se dividen entre segmentales y suprasegmentales, acorde a su estructura temporal. Las características prosódicas se califican del segundo tipo, y estas no segmentan la señal, sino que se aplican sobre la extensión entera. Al contrario que las características segmentales (temporales y espectrales) a las que se les aplica técnicas de *windowing* y se calculan una vez cada cierto segmento de tiempo (unos 20-50 mseg) [25].

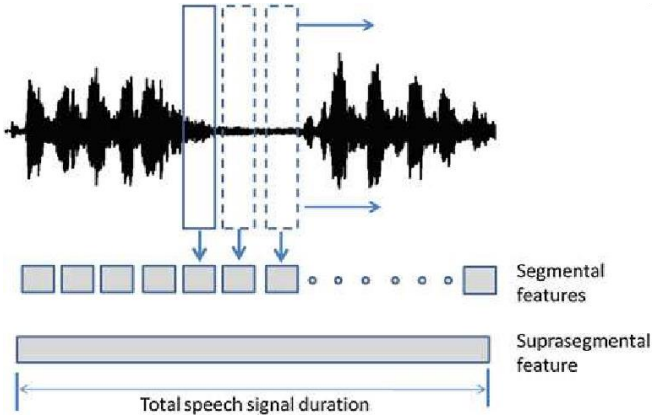


Figura 5 Características Segmentales y Suprasegmentales (Anagnostopoulos, Iliou & Giannoukos, 2012)

En estudios más tempranos, el foco se ponía principalmente en las características prosódicas, mientras que en investigación más reciente el uso de características espectrales ha crecido exponencialmente [25]. Escoger qué características (*features)* son las más útiles para el entrenamiento puede ser una tarea compleja [9]. Es importante que la selección de características no se vea afectada por la cultura, la región o el acento del locutor. Dar con la combinación de características acústicas más robusta para el reconocimiento automático de la emoción de un locutor es uno de los pilares del SER.

MFCC

Los Coeficientes Cepstrales en la Escala de Mel (MFCC), alias '*Most Frequently Considered Coeficcients'*, representan la amplitud del espectro del habla de manera compacta, esto los convierte en la técnica de extracción de características más usada en reconocimiento del habla. Cualquier sonido generado por el ser humano está determinado por la forma de su tracto vocal, por lo tanto, representar este sonido va a depender de poder determinar la forma correctamente [26]. Además, para lograr la mayor eficiencia del sistema de extracción de características, lo ideal sería imitar el comportamiento frecuencial del oído humano. De aquí surge el concepto de los coeficientes MFCC, basado en la escala de frecuencia MEL para imitar el comportamiento de tonos con distinta frecuencia dentro del oído humano.

SER: Feature Selection

La selección de características es el proceso en el que se identifica de manera automática o manual las características que más información aportan sobre la variable de predicción o salida. En muchos casos, las variables empleadas en el modelo final van a ser menos que las recopiladas desde el comienzo (figura 6).

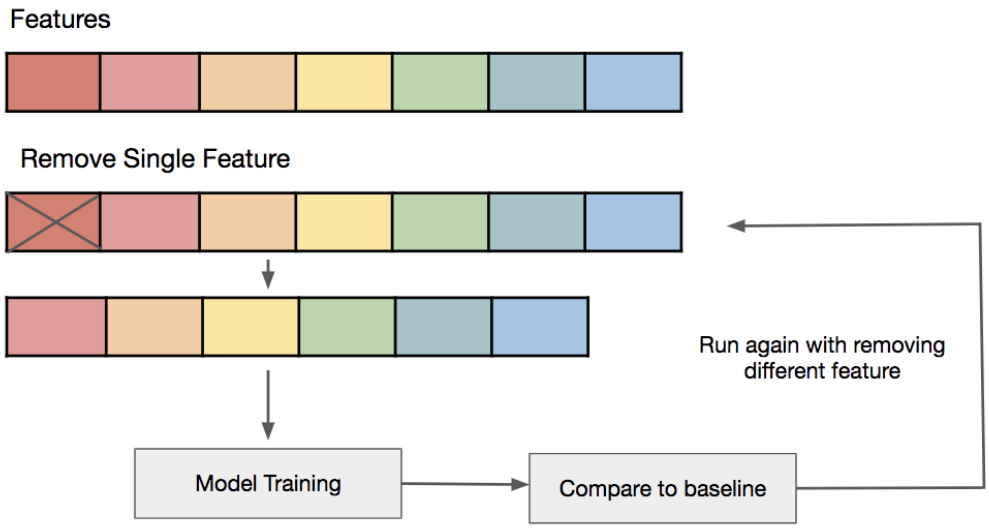


Figura 6 Esquema de Machine Learning con extracción de características (Chuan-En Lin, 2020)

La selección de características es uno de los pilares del aprendizaje automático que impacta de manera significativa al rendimiento del modelo. Los beneficios de esta práctica los resume la famosa frase "Menos es más" del arquitecto Ludwig Mies van der Rohe [27], impulsor del minimalismo. En aprendizaje automático, algunos de los principales beneficios de seleccionar la menor cantidad de atributos son [28]:

- Reduce la complejidad del modelo. Esto permite trabajar con un modelo más fácil de entender y que consuma menos recursos computacionales.

- Reduce el sobre entrenamiento. La probabilidad de tomar decisiones en torno al ruido de los datos se reduce al existir menos datos redundantes.
- Mejora la precisión. El modelo va a ser más eficaz al haber menos redundancia en los datos de entrenamiento
- Reduce el tiempo de entrenamiento. Los algoritmos aprenden más rápido al tener que entrenar con menos datos.

Los tipos de reducción de características vienen resumidos en la figura 7.

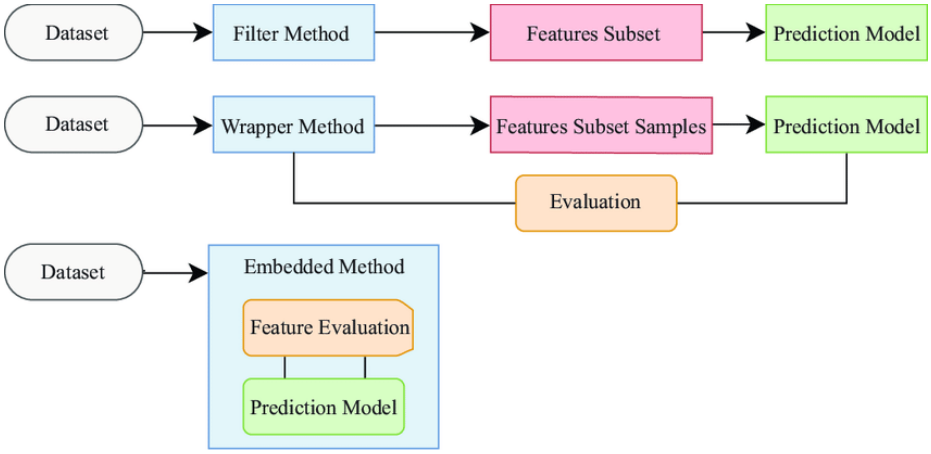


Figura 7 Principales métodos de selección de características en Machine Learning (Karaarslan, 2019)

El análisis de componentes principales (PCA) es una de las técnicas empleadas cuando se trabaja con datos de altas dimensiones [29]. PCA devuelve los componentes principales del conjunto de datos determinando la correlación entre características, por lo que se ha empleado como selector de características en numerosos estudios [30]. Al ser un método que permite reducir el tamaño de conjuntos de datos que incluyen una gran cantidad de características interrelacionadas, permite que los datos finales puedan constituir un menor número de variables [31]. Para este estudio el foco se pone en esta técnica, empleada como un *Filter Method* (figura 7).

SER: Feature Classification

Para la tarea de clasificación de características se emplean clasificadores lineales y no lineales. Los clasificadores lineales más usados incluyen las *Bayesian Networks* (BN) o el *Maximum Likelihood Principle* (MLP) y el *Support Vector Machine* (SVM). Sin embargo, la señal de voz se considera no estacionaria, por lo que un clasificador no lineal es óptimo para tareas SER [32]. Existen muchos clasificadores no lineales, los más usados en la detección de emociones por medio de voz son el *Gaussian Mixture Model* (GMM) y el *Hidden Markov Model* (HMM) [33].

Las investigaciones más recientes se han desarrollado en el terreno de los sistemas de aprendizaje profundo. Estos sistemas infieren una representación jerárquica de los datos de entrada, lo que facilita su categorización [34]. Estos sistemas forman parte de la rama de estudio conocida como el Deep Learning, cuya investigación y aplicación ha experimentado un fuerte auge durante los últimos años [35]. Las técnicas de Deep Learning tienen ciertas ventajas sobre los métodos tradicionales. Una de las más destacadas es su habilidad de detectar estructuras complejas de las características sin la necesidad de hacer una extracción manual [32].

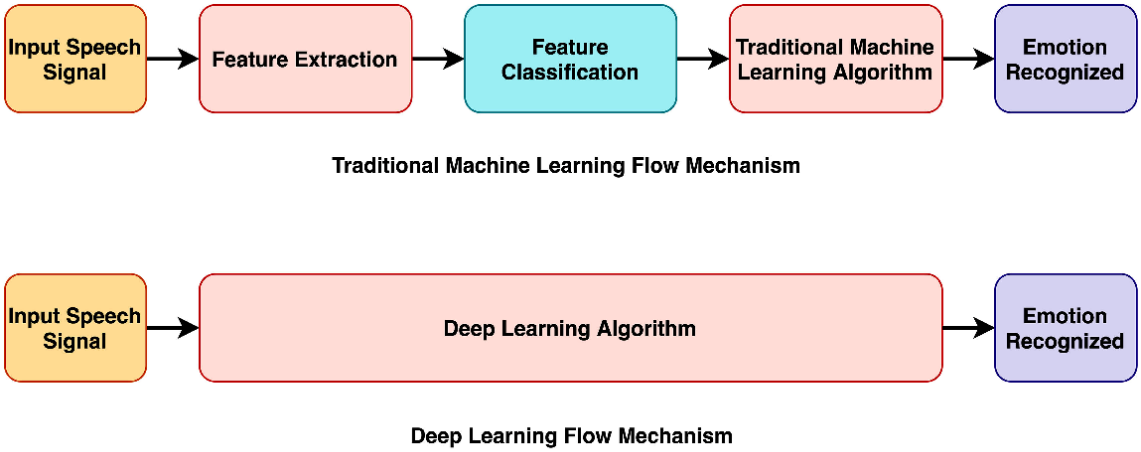


Figura 8 Método tradicional de Machine Learning vs. Deep Learning (Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H., & Alhussain, T. , 2019)

Las redes de aprendizaje profundo (DNNs) se basan en estructuras prealimentadas (*feed-forward*) compuestas por una o más capas escondidas subyacentes entre datos de entrada (*inputs*) y de salida (*outputs*), esta estructura se plasma en la figura 9. Las redes neuronales convolucionales (CNN) forman parte de este tipo de redes neuronales. Las redes *feed-forward* han demostrado mayor rendimiento en el procesamiento de video e imágenes.

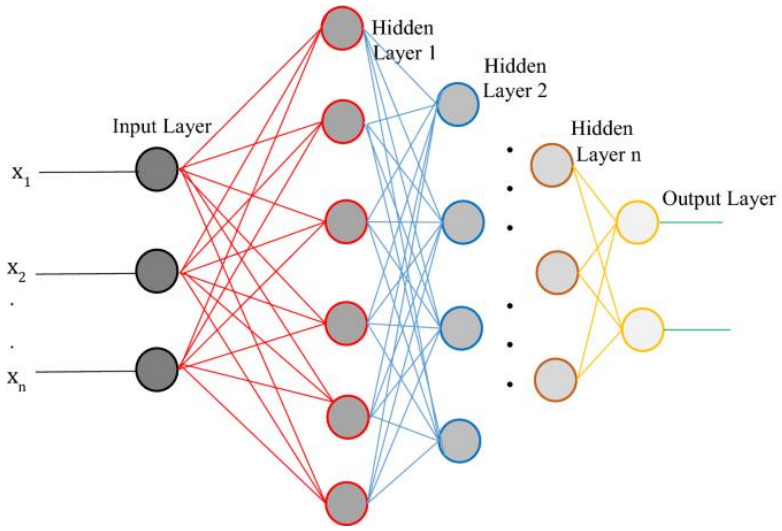


Figura 9 Arquitectura DNN (Glory, H.A., Vigneswaran, C., Jagtap, S.S. et al., 2021)

Por otro lado, están las arquitecturas recurrentes como las redes neuronales recurrentes (RNNs) y la memoria a corto largo plazo (LSTM), que han mostrado ser las más eficientes en tareas de procesamiento de lenguaje natural (NLP) [36]. Las RNN siguen el principio de retroalimentación, siguiendo un proceso que guarda la salida de una capa y la devuelve a la entrada, permitiendo que la información se reutilice y persista. Las CNN consideran solo la entrada actual, mientras que una RNN considera la entrada actual y también las entradas recibidas anteriormente, lo que le permite memorizar entradas anteriores gracias a su memoria interna [37].

Una red del tipo RNN se especializa en problemas de predicción de secuencias, como predecir cual sería la secuencia de palabras que alguien utilizaría en una búsqueda de Google. Sin embargo, uno de los retos a los que se enfrenta es agregar una nueva información sin perder de vista la información 'importante'. Aquí es donde entran en juego las LSTM, que hacen pequeñas modificaciones a la información mediante transformaciones multiplicativas y aditivas, de tal manera que la información fluye a través de un mecanismo conocido como 'estados de celda' (figura 10). De esta manera, los LSTM pueden recordar u olvidar cosas de forma selectiva ya que la información en un estado de celda particular tiene dependencias diferentes [38].

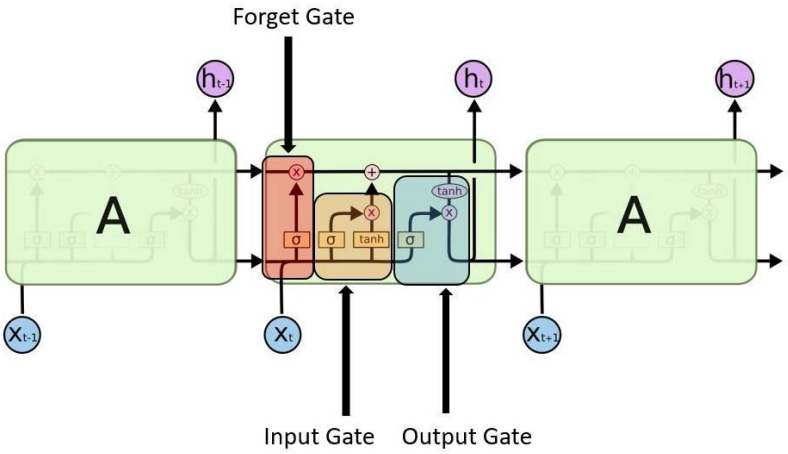


Figura 10 Celdas y estructura de LSTM (Mittidal, 2019)

Los estudios más recientes que han adoptados estos modelos son aquellos dirigidos a la implementación de sistemas de detección automatizada del Covid-19. Md. Zabirul Islam et al. (2020) aplicaron una técnica de aprendizaje profundo basada en la combinación de una red neuronal convolucional (CNN) y una memoria a largo-corto plazo (LSTM) para diagnosticar COVID-19 automáticamente a partir de imágenes de rayos X. En este sistema, CNN se usa para la extracción profunda de características y LSTM se usa para la detección del virus usando las características extraídas [39].

Aplicaciones web

Existen varios *frameworks* que permiten construir una página web usando Python como Bottle, Django o Flask. Este último es una opción popular para desplegar modelos de redes neuronales pre entrenados. Flask brinda más versatilidad para programar: es como un lienzo vacío para crear aplicaciones basadas en Python y tiene pocas dependencias. Flask es un *micro framework* basado en Werkzeug, el kit de herramientas WSGI y el motor Jinja2, que son proyectos de un grupo de programadores llamado Pocco [40].

- *Web Server Gateway Interface* (WSGI) se ha adoptado como estándar para el desarrollo de aplicaciones web Python. WSGI es una especificación para una interfaz universal entre el servidor web y las aplicaciones web [41].
- *Werkzeug* es un conjunto de herramientas WSGI implementadas por Flask como una de sus bases para gestionar solicitudes, objetos de respuesta y otras funciones [40].
- *Jinja2* es un motor de plantillas para Python que permite combinar este lenguaje con fuentes de datos y desarrollar páginas web dinámicas [40]. Flask se usa para el back-end, pero hace uso de este lenguaje de plantillas para crear HTML, XML u otros formatos de Markup que se devuelven al usuario a través de una solicitud HTTP [42].

Aplicaciones representativas

A continuación, se mencionan algunas aplicaciones y casos de uso similares a la herramienta propuesta en este proyecto.

Empath [43]

Programa desarrollado por Smartmedical Corp. Su algoritmo original identifica la emoción del usuario mediante el análisis de las propiedades físicas de su voz. Basado en decenas de miles de muestras de voz, detecta su ira, alegría, tristeza, calma y vigor. Sus servicios son ofrecidos a través de una página web propia, mostrada en la figura 11.

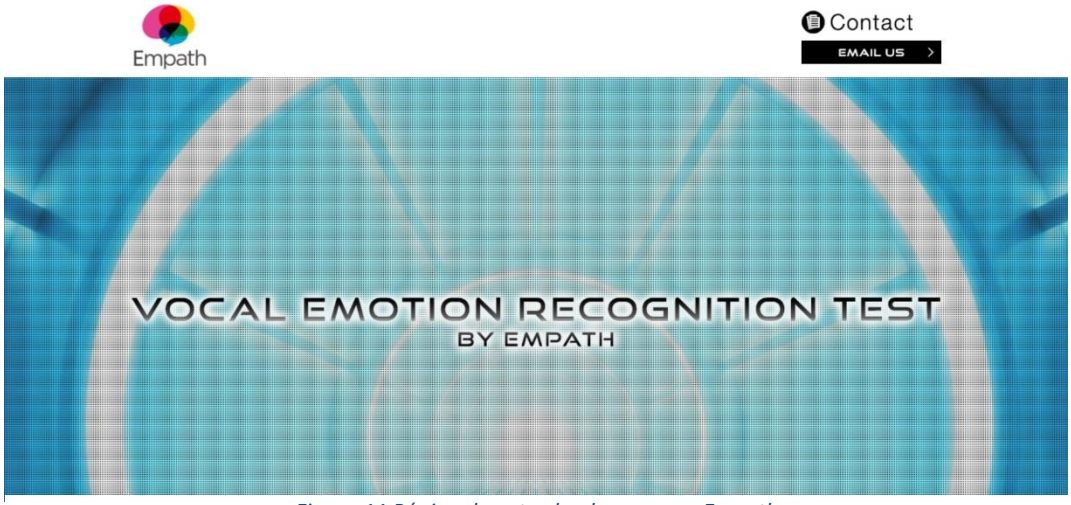


Figura 11 Página de entrada al programa Empath

Interview Simulator [44]

Plataforma de reconocimiento de emociones multimodal para analizar las emociones de los candidatos al empleo, en colaboración con la Agencia Francesa de Empleo. Se analizan las emociones faciales, vocales y textuales, utilizando principalmente enfoques basados en el aprendizaje profundo. Consiste en una aplicación web desplegada en Flask.

Figura 12 Página principal de la aplicación web Interview Simulator

Vmote [45]

Vmote es una aplicación de mensajería cuya interfaz de usuario está diseñada en torno a la voz. A través del Speech-to-Text permite enviar mensajes más rápido. Además, como los mensajes a menudo pueden malinterpretarse, Vmote también incorpora reconocimiento basado en emociones y detección de énfasis para proporcionar más contexto a los mensajes de voz.

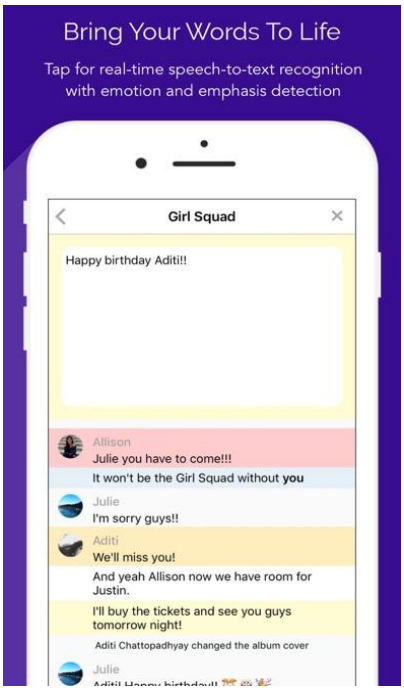


Figura 13 Plantilla de la UI de la app Vmote

2.2 Contexto y justificación

La medicina hipocrática defiende la necesidad de un estado de equilibrio y armonía entre el cuerpo y la mente. El padre y pionero de esta perspectiva de la medicina es Hipócrates (460-357 a.C.), una eminencia de la medicina de la Antigua Grecia [46]. Este mismo fue quien afirmó que para prevenir un ataque de asma, "el asmático debe protegerse de su propia ira" [46]. Hoy en día, más de 2 mil años después, se persigue una filosofía similar: "La salud es un estado de completo bienestar físico, mental y social y no solamente la ausencia de afecciones o enfermedades" [47], definición que proporcionó la Organización Mundial de la Salud en 1948. De ahí la aparición de la medicina holística, que defiende que el cuerpo, la mente y el ambiente contribuyen de manera equitativa a la salud [48].

¿Puede la mente curar el cuerpo? La corriente de pensamiento budista ha fomentado sus creencias sobre las capacidades curativas de la mente desde hace más de dos mil años [49], a lo que se han ido sumando cada vez más los científicos occidentales. Prueba de ello son numerosos encuentros entre el Dalai Lama y prominentes psicólogos, médicos y profesores de meditación, que han arrojado nueva luz sobre esta conexión cuerpo-mente [50].

Ya existen numerosos estudios que analizan el estado emocional y afectivo como factor determinante para el estado de salud de las personas. Cohen demostró que el estrés incrementa el riesgo de infecciones respiratorias agudas [51]. Con la aparición de la pandemia del Covid-19, es posible que un cuidado de la salud mental pueda fortalecer las defensas y ser un aliado a la hora de combatir esta enfermedad. Ejemplo de ello es un estudio que comprobó que mantener una actitud positiva durante momentos de elevado estrés promovía los niveles de S-Iga (inmunoglobulina-A salival), reconocida como una sustancia que protege contra las enfermedades respiratorias [52].

Si las emociones juegan un papel tan importante en la salud, fomentar un correcto estado de salud mental podría mejorar la salud de muchos ciudadanos. Especialmente ahora que vivimos en tiempos sin precedentes; la pandemia del Covid-19 ha impactado cada aspecto de nuestras vidas y nos ha llevado a una nueva realidad. Una realidad que está afectando a la salud tanto física como mental. La monitorización masiva de la salud de la población es crucial, para así poder garantizar el bienestar de los ciudadanos y ayudar a descongestionar los servicios sanitarios.

A su vez, vivimos en un mundo con mucha riqueza tecnológica. El auge de estas últimas amplía el horizonte de oportunidades y nos permite abordar problemas a través de soluciones menos tediosas, más rápidas y eficaces. Entonces, dado lo mucho que las máquinas se han integrado en nuestras vidas, ¿por qué no darles la oportunidad de que nos conozcan mejor? A muchos se les pueden venir a la cabeza historias terroríficas de ciencia ficción, resultado de un uso descontrolado. Sin embargo, sembrando las bases éticas, y con el control adecuado, el uso de las tecnologías avanzadas puede traer múltiples beneficios y mejoras a nuestro día a día. De la cuestión anterior, nace otra, ¿pueden las máquinas detectar nuestras emociones? La respuesta a estas dos cuestiones se plantea en el siguiente trabajo.

2.3 Planteamiento del problema

La crisis sanitaria producida por el Covid-19 ha multiplicado los sentimientos de miedo y preocupación y elevado los niveles de estrés en la población. Ante esta nueva y desafiante realidad es importante que cuidemos tanto nuestra salud física como mental. La vida cotidiana ha cambiado, el distanciamiento físico requerido para mantener un estilo de vida seguro ha desembocado en trabajar o estudiar desde casa, el desempleo temporal y la falta de contacto con seres queridos y amigos [55].

Un estudio sobre el análisis de sentimientos en España durante la pandemia del Covid-19 refleja el gran impacto que ésta ha causado en el estado emocional de muchos españoles. Esta crisis es un reto para los gobiernos e instituciones sanitarias, a medida que los sentimientos de miedo, estrés y disgusto se propagan [56].

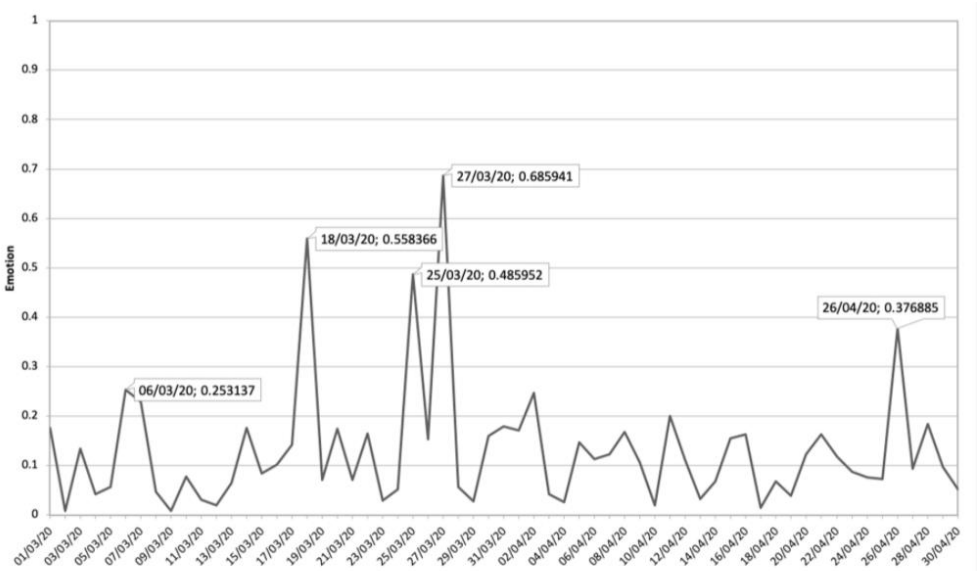


Figura 14 Evolución del sentimiento del miedo en España durante el periodo del 1 del marzo 2020 hasta el 30 de abril 2020 (de Las Heras-Pedrosa C, Sánchez-Núñez P, Peláez JI., 2020)

Se distinguen varios picos en los datos que coinciden con fechas específicas de ciertos titulares en los medios de comunicación. La figura 15 detalla la difusión de información en ciertos días del confinamiento que coinciden con los picos observados en la figura 14. La OMS hizo un llamamiento en la Declaración sobre la reunión realizada el 29 de octubre de 2020 a la necesidad de fortalecer los servicios de salud mental ante la crisis sanitaria del Covid-19 [57]. La monitorización de las emociones a escala nacional puede ayudar a muchos gobiernos a gestionar sus políticas de actuación.

	Fear Emotion Related to COVID-19 Comments in Digital Ecosystems
6 March 2020	• Loss of jobs.
18 March 2020	• The virus preys on residences. • Closing of borders.
25 March 2020	• Spain surpasses China in coronavirus deaths with 3434 deaths. • The number of deaths continues to rise. • The Spanish government now estimates more than three months of economic collapse. • Collapsed health system.
27 March 2020	• The virus has reached the level of a pandemic. • The Spanish government is overwhelmed.
26 April 2020	• Censorship by the Spanish Government. • Increased unemployment. • Defective medical equipment. • Increased unemployment. • Defective medical equipment.

Figura 15 Noticias relacionadas con el miedo y el Covid-19 en los medios de comunicación (de Las Heras-Pedrosa C, Sánchez-Núñez P, Peláez JI., 2020)

Mucha gente tiene que luchar sola contra estas situaciones emocionalmente complejas, lo cual puede suponer un empeoramiento de la salud mental y física. La población de avanzada edad es especialmente vulnerable; para muchos tener que asistir a la consulta de un médico de manera física supone un alto riesgo, muchas veces vital. Por lo tanto, se considera importante poder ofrecer alternativas al contacto físico que puedan proporcionar seguridad en momentos de mayor malestar emocional.

Uno de los mayores problemas a los que se enfrenta la psicología es la dificultad de diagnóstico de ciertos trastornos, por lo que poder integrar un sistema de captación de emociones en el día a día de las personas, sobre todo aquellas más propensas a poder sufrir alguno de estos trastornos, puede ser una vía de detección temprana de enfermedades. Esta herramienta, por lo tanto, puede mejorar del estado de salud de muchas personas.

Sin embargo, también se le puede dar una utilidad más casual y lúdica: “¿Qué emociones estoy transmitiendo a los demás?”. Poder detectar y monitorizar patrones en la voz puede ayudar a entender y gestionar la forma en la que nos comunicamos en el día a día. El seguimiento de estas emociones puede, a su vez, fortalecer la conciencia propia.

El principal problema que afronta el AC según Picard es que los perfiles de cada persona son muy particulares, lo que convierte la tarea predecir los estados afectivos en una con una alta variabilidad [12]. Por lo tanto, este estudio plantea el uso de características extraídas de la voz para mejorar la comprensión del estado afectivo de los usuarios. La voz se considera un input adecuado ya que toma diferentes formas para distintas emociones [12].

Una tarea indispensable es que la aplicación aprenda de manera autónoma. En este caso, se busca la estrategia de aprendizaje óptima a través de un algoritmo de clasificación que dé los mejores resultados en las fases de entrenamiento y predicción y empleando previamente las mejores prácticas de preparado de datos.

Capítulo 3. OBJETIVOS

3.1 Objetivos generales

El objetivo del presente trabajo es el desarrollo de un sistema que, basado en inteligencia artificial que pueda detectar el estado emocional del usuario. Además de desarrollar una aplicación web que a través de un sistema de reconocimiento de voz y pueda devolver esta predicción al usuario en tiempo real.

3.2 Objetivos específicos

La analítica de datos juega un papel importante en este proyecto. Para ello, se diseña un proceso ETL sobre el dataset previo a la implementación del algoritmo. Este proceso se desglosa en las siguientes transformaciones:

1. Extract:
   Encontrar y extraer los conjuntos de datos adecuados para el entrenamiento del algoritmo. Para la búsqueda de fuentes de datos adecuadas se va a utilizar la plataforma Kaggle, donde se encuentra una de las comunidades de científicos de datos más destacadas a nivel mundial.
2. Transform:
   - Solucionar los problemas de calidad de los datos en el caso de que los hubiese
   - Tratado de datos para su correcta ingesta por el algoritmo.
   - En el caso de tareas de clasificación del habla, se requiere una extracción de características (*Feature Extraction*) adecuada. Por lo tanto, requiere estudiar las opciones de extracción de características.
   - Seleccionar las características que optimizan el aprendizaje del algoritmo.
3. Load:
   Cargar los datos finales transformados en un lugar centralizado. Una vez estén los datos preparados para su uso, podrán ser empleados para entrenar el algoritmo.

A continuación, se especifica la parte técnica de inteligencia artificial:

4. Estudiar las opciones de algoritmos de sistemas de reconocimiento de voz.
5. Diseñar el sistema de reconocimiento de voz: la idea principal consistirá en un algoritmo de clasificación, en concreto una red neuronal artificial.
6. Decidir cuáles serán las métricas más significativas para escoger el modelo óptimo.
7. Análisis de los resultados obtenidos para cada iteración; realizar los ajustes necesarios en los parámetros, con el fin de escoger el mejor modelo.
8. Escoger el algoritmo con mejores resultados, capaz de predecir la emoción en datos nuevos.

La importancia de que el modelo muestre destreza a la hora de completar e interpretar estas tareas correctamente va a ser determinante a la hora de proporcionar las conclusiones adecuadas. Una vez el algoritmo se haya entrenado, se pasa a la fase de desarrollo web.

El dispositivo debe ser capaz de responder al usuario. Para ello, los principales sistemas integrados serán una serie de habilidades:

9. Captar los datos necesarios a través del micrófono
10. Preprocesamiento de los datos; repetir transformaciones especificadas anteriormente
11. Analizar y dar respuesta a los datos

Se debe implementar una interfaz de usuario adecuada. Este proyecto propone una aplicación web desarrollada con un *framework* adecuado a Python. Los objetivos específicos que se persiguen para este bloque son:

1. Estudiar alternativas para el desarrollo de una aplicación web que sea capaz de reconocer las emociones en la voz de un usuario y devolver un resultado en tiempo real.
2. Determinar cómo integrar el sistema de predicciones con la estructura de la aplicación.
3. Diseño de la interfaz de usuario. Esto incluye el diseño de un logo.
4. Incorporar permisos de grabación de voz y un sistema de grabación de audio en tiempo real.
5. Conectar todos los componentes a través de solicitudes HTTP adecuadas.

3.3 Beneficios del proyecto

Con este sistema un usuario independiente puede mantener una monitorización del estado de sus emociones e impulsar una rutina hacia una vida más saludable, o gestionar la forma en la que se comunica y toma decisiones, sin necesidad de desplazamientos. Muchos comportamientos del ser humano se ven condicionados por las las emociones (recuerdos, memoria, aprendizaje, toma de decisiones…). Determinar el estado emocional de las personas puede ser un elemento de ayuda para detectar enfermedades (punto de vista clínico), por lo tanto, un sistema de detección de emociones riguroso puede formar parte del diagnóstico de enfermedades del sistema nervioso.

A su vez, el sistema de reconocimiento de emociones puede ser utilizado por empresas en el campo de la inteligencia artificial y desarrollo tecnológico para reforzar sus investigaciones sobre el campo SER, NLP y otras aplicaciones dentro de la dinámica comunicación hombre-maquina. A su vez, puede ser un jugador importante dentro del neuromarketing.

La herramienta fundamental para poder usar el producto es un ordenador, de uso general, por lo que es un producto bastante accesible. La interfaz desarrollada sirve como prototipo del funcionamiento del sistema de reconocimiento de emociones y como manera de compartir el motor principal de inteligencia artificial de manera más visual.

DESARROLLO DEL PROYECTO

3.4 Planificación del proyecto

Fase 1. Investigación previa

1. Estudio del del contexto y el estado del arte de los conceptos y tecnologías del proyecto. Esto incluye la búsqueda de las alternativas y casos de uso actuales.
2. Análisis técnico de las tecnologías escogidas, tanto para el componente back-end y el front-end.

Fase 2. Planificación y diseño

1. Definir los objetivos principales del proyecto.
2. Determinar el alcance de los objetivos y la distribución de tiempo correspondiente.
3. Definición de requisitos y recursos necesarios. Estos pueden ser requisitos tanto software como hardware.
4. Diseño de la arquitectura back-end del sistema: preparado de datos, extracción de características y en el algoritmo de reconocimiento de emociones y sus predicciones.
5. Diseño de la arquitectura front-end del sistema. Esta será una aplicación web capaz de captar la voz del usuario, enviarla a donde se aloje el algoritmo, y recibir una predicción en tiempo real.
6. Diseño del proceso de validación y pruebas del sistema

Fase 3. Preparación del entorno

1. Reunir los requisitos necesarios para preparar el entorno de trabajo. Aplicaciones, programas y elementos como SDKs necesarios para el correcto funcionamiento en la maquina local.
2. Empezar a manejar aquellas tecnologías que son nuevas y no se hayan usado previamente al desarrollo del proyecto. Para ello se emplean videos y cursos online relacionados con estas competencias.

Fase 4. Desarrollo e implantación de la arquitectura

ETL:

1. Buscar los datasets adecuados para el entrenamiento. Los datos deben tener un formato parecido o igual y con un labelling adecuado.
2. Desarrollar un pipeline adecuado de ingesta y preparado/limpieza de datos. Incluye determinar las librerías necesarias para ello.
3. Llevar a cabo un análisis exploratorio previo de los datos. Se pretende encontrar patrones o indicios de fenómenos como el desbalance de clases.

Desarrollo del clasificador SER:

4. Estudiar las opciones de algoritmos para la tarea de clasificación de emociones. Plantear una serie de pruebas con aquellos considerados más adecuados.
5. Determinar la estrategia de extracción de características de los datos y prepararlos para poder ser ingeridos por el algoritmo elegido. Para ello, se investigarán tareas de escalado de datos y separación entre entrenamiento, validación y test.
6. Iniciar la fase de entrenamiento. Probar con diferentes valores para los hiperparámetros de los modelos y fijar las métricas que se utilizarán para la evaluación.
7. Comparativa de modelos y escoger en base a la eficacia de predicción.

Interfaz de usuario:

1. Estudiar alternativas para el desarrollo de una aplicación web que sea capaz de reconocer las emociones en la voz de un usuario y devolver un resultado en tiempo real.
2. Determinar cómo integrar el sistema de predicciones con la estructura de la aplicación.
3. Diseño de la interfaz de usuario. Esto incluye el diseño de un logo.
4. Incorporar permisos de grabación de voz y un sistema de grabación de audio en tiempo real.
5. Conectar todos los componentes a través de solicitudes HTTP adecuadas.

Fase 5. Pruebas, conclusiones y análisis de rendimiento

1. Periodo de pruebas, refuerzo e identificación de disfunciones. Se deberán aplicar los parches y las medidas necesarias para arreglar posibles fallos de funcionamiento.
2. Revisión de protocolos de seguridad de datos (RGPD, LSSI) en caso de que fuesen necesarios.
3. Redactar la memoria a partir de todos los documentos empleados y redactados durante cada fase de desarrollo.
4. Defensa del TFG.

Nombre de la tarea	Mes						
	Enero	Febrero	Marzo	Abril	Mayo	Junio	N.º horas
Investigación previa							
Contexto y estado del arte							20
Análisis técnico de las tecnologías empleadas							
Planificación y diseño							
Definir objetivos y el alcance							30
Definir requisitos y recursos							
Diseño de la arquitectura back-end							
Diseño de la arquitectura front-end							
Diseño del proceso de pruebas							
Preparación del entorno							
Instalar y configurar los requisitos computacionales necesarios							90
Aprendizaje de nuevas tecnologías							
Reunión de seguimiento con la tutora del TFG							
Desarrollo e implantación							
ETL							

Búsqueda de datasets para el entrenamiento																								50
Desarrollo del pipeline de ingesta y preparado/limpieza de datos																								
Análisis exploratorio de los datos limpios																								
Reunión de seguimiento con la tutora del TFG																								
<u>Desarrollo del Clasificador SER</u>																								
Estudiar algoritmos de clasificación																								150
Extracción de características																								
Reunión de seguimiento con la tutora del TFG																								
Entrenamiento del modelo y ajuste de hiperparémetros																								
Reunión de seguimiento con la tutora del TFG																								
Comparativa y elección del mejor modelo																								
<u>Interfaz de usuario</u>																								
Estudiar alternativas para el desarrollo de una aplicación web																								
Determinar cómo integrar el sistema de predicciones																								

con la estructura de la aplicación																									130
Reunión de seguimiento con la tutora del TFG. Revisión de los algoritmos.																									
Diseño de la interfaz de usuario																									
Incorporar sistema de grabación de audio en tiempo real y conectar todos los componentes a través de solicitudes HTTP adecuadas																									
Reunión de seguimiento con la tutora del TFG																									
Pruebas, conclusiones y análisis de rendimiento																									
Pruebas y ajustes de funcionamiento																									100
Políticas de seguridad de datos																									
Redacción de la memoria final																									
Entrega primera versión completa de la memoria de TFG a la tutora																									
Realizar cambios sugeridos en el feedback de la tutora de TFG																									
Entrega de la versión final de la memoria de TFG																									
DEFENSA DEL TFG																									

3.5 Descripción de la solución, metodologías y herramientas empleadas

El proyecto está dividido en dos grandes componentes. Primero se desarrolla un sistema de aprendizaje automático motorizado por un pipeline de procesado de datos, extracción de características e implementación de un modelo entrenado y capaz de realizar predicciones a partir de nuevas fuentes de datos. Esta primera fase se muestra en la figura 16.

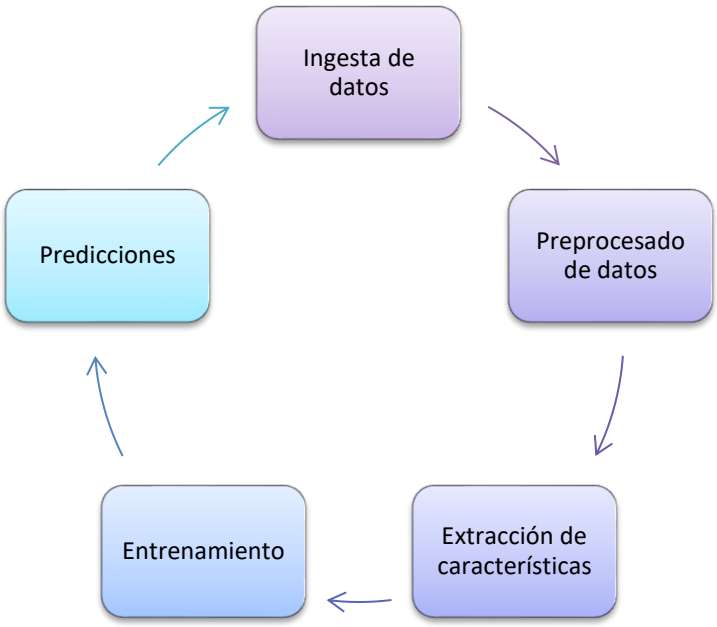


Figura 16 Pipeline principal detrás del sistema de reconocimiento de voz

La segunda parte consiste en el desarrollo de una aplicación web como aplicación práctica al sistema de detección de emociones desarrollado.

3.5.1 Fuentes de datos

El entrenamiento del algoritmo de clasificación de emociones es el núcleo del presente trabajo. Se decide combinar dos datasets diferentes para poder tener un conjunto de entrenamiento grande y variado, pero evitando mezclar datos de diferentes formatos ya que puede dar lugar a incongruencias en el entrenamiento. A su vez, ambos datasets contienen voces masculinas y femeninas, lo cual permite desarrollar algoritmos especializados para cada género y uno mixto. Los datasets elegidos son los siguientes:

- **RAVDESS** (Ryerson Audio-Visual Database of Emotional Speech and Song) (Livingston & Russo, 2018), datos publicados bajo una licencia de atribución de Creative Commons de uso no comercial [58].

La RAVDESS contiene 7356 archivos, en los que participan 24 actores profesionales (12 mujeres, 12 hombres) con un acento norteamericano neutral. El habla incluye expresiones de calma, felicidad, tristeza, ira, miedo, sorpresa y disgusto, y se vocaliza una de dos frases. La canción contiene emociones tranquilas, felices, tristes, enfadadas y temerosas. Cada expresión tiene dos niveles de intensidad emocional (normal, fuerte), con una expresión neutra adicional.

Para este proyecto solo se van a usar archivos de audio y de tipo voz. El archivo de voz (Audio_Speech_Actors_01-24.zip, 215 MB) contiene 1440 archivos: 60 ensayos por actor x 24 actores = 1440. La tabla 3 detalla las características de los archivos.

Nº	Característica	Opciones
1	Modalidad	01 = AV completo, 02 = solo video, 03 = solo audio
2	Canal vocal	01 = habla, 02 = canción
3	Emoción	01 = neutral, 02 = calma, 03 = feliz, 04 = triste, 05 = enojado, 06 = temeroso, 07 = disgusto, 08 = sorprendido
4	Intensidad emocional	01 = normal, 02 = fuerte (no hay intensidad fuerte para la emoción neutral)
5	Frase	01 = “Kids are talking by the door”, 02 = “Dogs are sitting by the door”
6	Repetición	01 = 1ª repetición, 02 = 2ª repetición
7	Actor	01 a 24. Los actores impares son hombres, los actores pares son mujeres

Tabla 3 Descripción de la identificación de archivos de audio de la base de datos RAVDESS

Ejemplo de nombre de archivo: 02-01-06-01-02-01-12.mp4

- Solo video (02)
- Discurso (01)

- Temeroso (06)
- Intensidad normal (01)
- Declaración "perros" (02)
- 1a repetición (01)
- 12 ° actor (12)
- Mujer, ya que el número de identificación del actor es par.

- **CREMA-D** (Crowd-sourced Emotional Multimodal Actors Dataset) [59] es un conjunto de datos de 7,442 archivos de 91 actores. Las voces las componen 48 actores masculinos y 43 femeninos entre las edades de 20 y 74, provenientes de una variedad de razas y etnias (afroamericanos, asiáticos, caucásicos, hispanos y no especificados).

  Los actores hablan una selección de 12 frases expresadas en una de seis emociones diferentes (ira, disgusto, miedo, feliz, neutral y triste) y cuatro niveles de emoción diferentes (bajo, medio, alto y no especificado).

Nº	Característica	Opciones
1	ID del actor	Número de 4 dígitos. Los IDs 1002-1013,1018,1020,1021,1024,1025,1028-1030,1037,1043,1046,1047,1049, 1052-1056,1058,1060,1061,1063,1072-1076,1078,1079,1082,1084,1089,1091 corresponden al sexo femenino, y el resto al masculino.
2	Frase	“It's eleven o'clock” (IEO). “That is exactly what happened” (TIE). “I'm on my way to the meeting” (IOM). “I wonder what this is about” (IWW). “The airplane is almost full” (TAI). “Maybe tomorrow it will be cold” (MTI). “I would like a new alarm clock” (IWL) “I think I have a doctor's appointment” (ITH). “Don't forget a jacket” (DFA). “I think I've seen this before” (ITS). “The surface is slick” (TSI).

		“We'll stop in a couple of minutes” (WSI).
3	Emoción	Enfado (ANG) Disgusto (DIS) Miedo (FEA) Feliz (HAP) Neutro (NEU) Triste (SAD)
4	Nivel de emoción	Bajo (LO) Medio (MD) Alto (HI) Sin especificar (XX)

Tabla 4 Descripción de la identificación de archivos de audio de la base de datos CREMA-D

Ejemplo de nombre de archivo: 1001_DFA_ANG_XX.wav

- ID del actor (1001), actor masculino
- Frase “Don’t forget a jacket” (DFA)
- Enojado (XX)
- Nivel de emoción no especificado (XX)

3.5.2 Ingesta, limpieza y análisis exploratorio de datos

Para los procesos de tratado de datos y entrenamiento se implementa una arquitectura en el entorno de trabajo Jupyter Notebook dentro de la distribución Anaconda. Esta distribución es de especial interés para este trabajo ya que se manejan paquetes que involucran complejas redes de dependencias (especialmente Tensorflow y Keras) que pueden resultar especialmente tediosas si se trabaja en sistemas Windows. Anaconda facilita la configuración del entorno de trabajo necesario para este proyecto, que se desarrolla dentro de un *virtual environment* que encapsula todas las librerías necesarias.

Una vez importados todos los datos, se organizan de tal manera que los datos estén etiquetados con su emoción correspondiente para que el clasificador pueda aprender y diferenciar cada emoción. Al estar trabajando con dos datasets diferentes, se crea un dataframe de *pandas*, donde se almacenan todos los datos con sus respetivas etiquetas. Este dataframe es el que se va a emplear para las próximas fases de preprocesamiento de datos y de extracción de características.

En un modelo predictivo de aprendizaje automático las tareas de automatización se ven muy influenciadas por la calidad de los datos y el nivel de representatividad del grupo de entrenamiento [60]. La limpieza de datos es un proceso fundamental en toda tarea de machine learning. Permite eliminar errores y datos redundantes para lograr un dataset más fiable [61] lo cual va a mejorar la calidad de entrenamiento del clasificador. Sin embargo, los datasets escogidos no necesitan manipulación, al tratarse de fuentes de datos de alta difusión, por lo que no es necesario ningún tipo de transformación para eliminar, modificar o sustituir datos.

Sin embargo, se detecta un problema de desbalance de clases. Es un desafío para las tareas de modelado predictivo trabajar con una distribución de clases severamente sesgada, causando desigualdad en los costos de la clasificación errónea [62]. De tal manera, el algoritmo se obstruye prediciendo las clases mayoritarias y no aprenda sobre las más pequeñas. Una solución es eliminar estas categorías de datos, pero en este proyecto se consideran las ocho emociones igual de importantes, y un aspecto considerado importante es añadir variabilidad al modelo, dotándole de la capacidad de predecir un amplio abanico de emociones, el cual se pretende agrandar en futuras líneas de trabajo. Para problemas en los que estas categorías más escasas son importantes, se aplican técnicas de *oversampling* o *undersampling* [63]. La primera consiste en generar más datos de las clases minoritarias y la segunda en reducir la cantidad de datos de las clases mayoritarias. En este caso, se aplica *oversampling*, ya que se considera importante trabajar con cantidades de datos elevadas, así el algoritmo tiene más información para el aprendizaje, razón por la que además de hacer un *oversampling* de las clases de 'calm' y 'surprise', también se hace del resto de clases. De tal manera, se obtiene un dataset con cuantiosos datos y con equilibrio entre clases.

Uno de los enfoques más empleados para sintetizar datos nuevos es el Synthetic Minority Oversampling Technique (SMOTE) [64]. Esta técnica lo que hace es seleccionar muestras aleatorias de la clase minoritaria y usa un algoritmo KNN para seleccionar vecinos a los que se dibujan líneas [65].

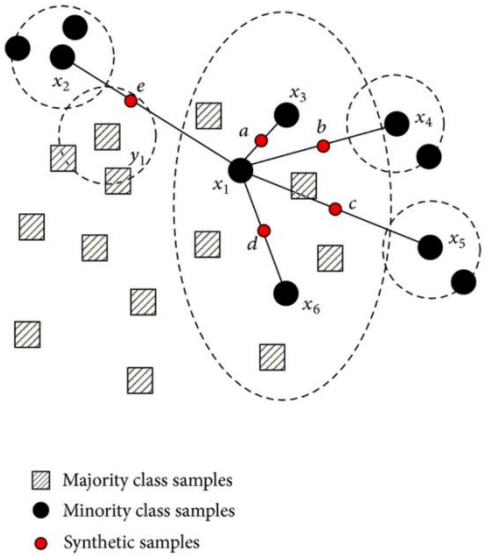


Figura 17 Representación del funcionamiento detrás de la técnica de oversampling SMOTE (Indresh Bhattacharyya, 2018)

Una vez está preparado el conjunto de datos, ya se pueden realizan las operaciones necesarias sobre todo el conjunto.

3.5.3 Generación de datos artificiales (Data Augmentation)

Las técnicas de Data Augmentation se utilizan para crear nuevos datos sintéticos de entrenamiento añadiendo pequeñas perturbaciones a los datos iniciales. Esto da lugar a una mayor precisión y a una mayor generalización, ya que el objetivo es hacer al modelo invariante a las posibles perturbaciones que pueden encontrarse en muestras de audio [66]. Cuando se trata de una tarea con archivos de audio, se pueden generar datos artificiales con las técnicas especificadas en la tabla 5 [67].

Técnica	**Descripción**	**Librería de Python empleada**
Noise injection	Elevar el nivel de ruido a partir de un número aleatorio	Numpy
Stretch	Transformar la señal utilizando la Transformada de Fourier de Tiempo	Librosa

	Reducido (STFT), la estira usando un codificador y usa la inversa de la STFT para reconstruir la señal en el dominio de tiempo	
Shift	Cambiar a el audio hacia la: - izquierda (avance rápido) x segundos, lo que marca los primeros x segundos como 0 - derecha (rebobinar) x segundos, los últimos x segundos se marcarán como 0	Librosa
Pitch	Cambiar el tono al azar	Librosa
Speed	Extiende o disminuye las series de tiempo una tasa fija	Librosa

Tabla 5 Técnicas de Data Augmentation, descripción y librería de Python utilizada para su aplicación

3.5.4 Extracción de características (Feature Extraction)

Para este proyecto, se aplican los beneficios de las características MFCC. Esta va a ser la técnica empleada, ya que se considera que estas características aportan un conocimiento suficientemente alto a los modelos sobre los audios de entrenamiento. Realizar una extracción de características empleando muchas técnicas resulta distorsionar la información contenida en los audios.

Para extraer los coeficientes MFCC se emplea una vez más la librería *librosa* que ofrece una función para ello y en la que se pueden ajustar distintos parámetros para adecuarla al formato de audio con el que se trabaja. La figura 18 muestra el proceso detrás de la elaboración de un vector de MFCC.

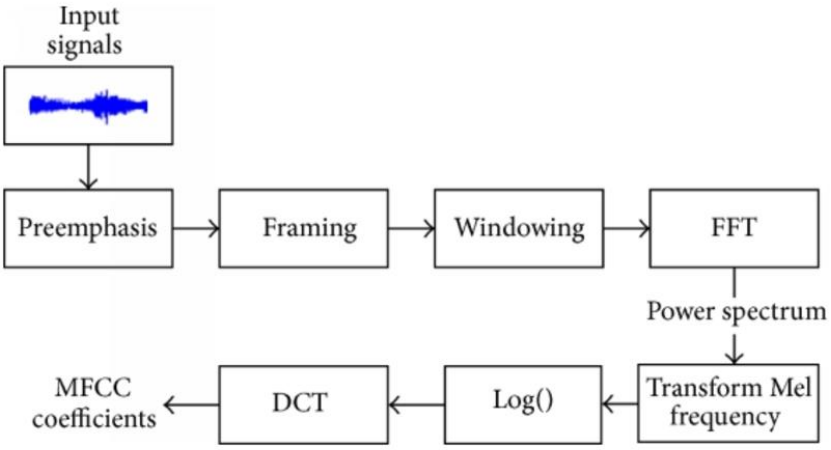


Figura 18 Proceso de extracción de los coeficientes MFCC durante el Feature Extraction (Gong, S., Dai, Y., Ji, J., Wang, J., & Sun, H., 2015)

- Primero se aplica un filtro de pre-énfasis que enfatiza las frecuencias de resonancia de la cavidad acústica del tracto vocal.
- Posteriormente se divide la señal en tramos (*framing*) y se le aplica una función de *windowing*, generalmente se trabaja con una ventana de Hamming. El *windowing* sirve para eliminar los bordes de la señal y acentuar la parte central para su análisis.
- Al obtener la Transformada Rápida de Fourier (FFT) se extrae la magnitud de frecuencia de cada tramo y esta información se pasa a escala de Mel mediante el Banco de Filtros (Transform Mel frecuency). MFCC utiliza la escala Mel, que se divide en filtros espaciados linealmente a baja frecuencia (debajo de 1000Hz) y logarítmicos por encima de 1000Hz. Por ello existen más filtros en zonas de baja frecuencia que en las de alta (figura 19) [68].

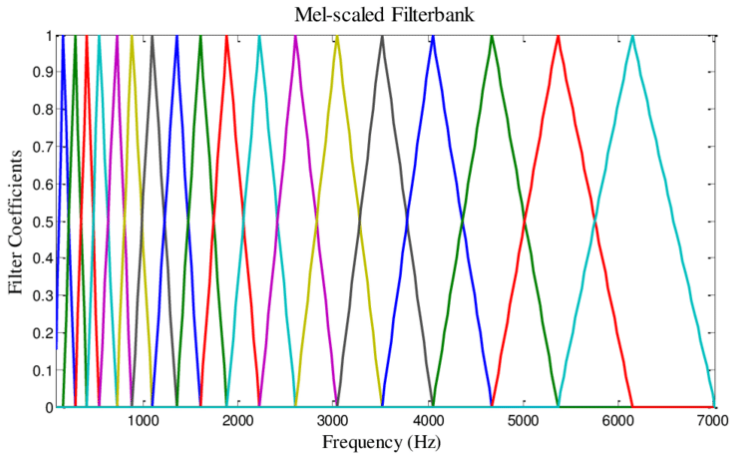


Figura 19 Mel filter banks basis functions using 20 Mel-filters in the filter bank (Yusnita, M. A., Paulraj, M. P., Yaacob, S., Yusuf, R., & Shahriman, A. B., 2013)

- Después se obtiene el logaritmo de la señal, lo que hace que las estimaciones de frecuencia no sean tan sensibles a variaciones en la señal.
- Finalmente se aplica la Transformada de Coseno Discreta (DCT) a los coeficientes del espectro mel para convertirlos al dominio del tiempo y obtener los coeficientes cepstrales (MFCC). Esta transformación viene dada por la fórmula:

$$y_t[k] = \sum_{m=1}^{M} \log(|Y_t(m)|^2)\cos\left[k * (m - 0.5) * \frac{\pi}{M}\right]$$

$$k = 1, 2, \ldots, J$$

Figura 20 Formula de la Transformada de Coseno Discreta (DCT)

M = Nº filtros Mel

J = número de MFCC

Se decide extraer 50 MFFCs por audio, ya que la tarea de reconocimiento de emociones es compleja. Al estar usando tan solo una técnica de extracción de características, se explota la máxima riqueza de estos coeficientes.

Este proceso se aplica por un lado a los audios de voz masculina y por otro a los de voz femenina, por lo tanto, el resultado final son dos datasets que comparten la misma estructura: una columna con el nombre de la etiqueta, y el resto son 50 columnas con los 50 coeficientes extraídos para cada audio.

3.5.5 Modelo y Entrenamiento

Preparado de datos

Una vez tenemos los datos en formato adecuando, se deben normalizar y dividir entre conjunto de entrenamiento y de pruebas. El proceso previo al entrenamiento se detalla en la figura 21.

Figura 21 Secuencia del preprocesamiento de datos

Figura 15.

Al tratarse de un problema de clasificación multi clase y que los datos son categóricos, el modelo se confunde si no se etiquetan adecuadamente. Por eso se aplica *One-Hot Encoding* a la etiqueta Y (las emociones), lo que genera una columna para cada valor distinto de las emociones que se están codificando y para cada registro, marca con un 1 la columna a la que pertenezca dicho registro y deja las demás con 0. Este es el formato adecuado para el entrenamiento [69].

Los datos se dividen en conjuntos de entrenamiento y test (80%-20%), repartiendo las emociones en cada grupo de manera aleatoria. Las dimensiones de los conjuntos de datos en este punto se muestran en la tabla 6.

Conjunto de datos	**Dimensiones**
Ambos generos	
x_train y_train	(68107,58) (68107,58)
x_test y_test	(17027, 58) (17027, 8)
Femenino	
x_train y_train	(39379, 58) (39379, 8)
x_test y_test	(9845, 58) (9845, 58)
Masculino	
x_train y_train	(28728, 58) (28728, 8)
x_test y_test	(7182, 58) (7182, 8)

Tabla 6 Dimensiones de los diferentes conjuntos de datos utilizados

Para el escalado de datos, se utiliza el módulo **StandarScaler()**, que escala los datos siguiendo una distribución normal (distribución Gaussiana con media 0 y varianza unitaria):

$$.Z = (X - \mu)/\sigma$$

Al trabajar con modelos de Deep Learning, si las características no siguen esta distribución, el modelo puede resultar inestable. Esto afecta de manera negativa al aprendizaje y a la sensibilidad hacia los valores entrantes, desembocando en un error de generalización mayor [62]. Para trabajar con los modelos descritos a continuación, también es necesario expandir la dimensión (+1) de los datos de entrada, para ello se utiliza la librería *numpy*.

Feature selection: PCA

La selección de características se realiza durante el preparado de datos para el modelo ya que es necesario realizar las transformaciones hasta este punto para poder aplicar PCA. Esta técnica permite comprobar si el número de coeficientes elegidos resulta tener dimensiones útiles para el modelo y poder tomar decisiones en torno a la posible necesidad de reducción de ruido.

En concreto se emplea la función de *explained variance* para cada componente. Esto es la razón entre la varianza de ese componente principal y la varianza total. Los gráficos mostrados en el apartado de resultados muestran la *explained variance* acumulada, lo que permite visualizar a partir de que característica la información aportada resulta redundante.

Durante las primeras pruebas se extrajeron las características ZCR, Chroma Shift, MFCC, RMSV, Mel Spectogram, las cuales formaban un total de 100 componentes. Al aplicar PCA, la curva de varianza acumulada indica que se estaba trabajando con características de más, una conclusión que fue enfatizada por los resultados de los entrenamientos mostrando un severo *overfitting*. Para el modelo final se decide extraer solamente coeficientes MFCC ya que se considera que contienen información suficiente sobre los datos, algo que se retrata en los exitosos resultados finales (tablas 11, 12, 13).

Arquitectura del modelo: CNN+LSTM

En cuanto a que tipo de red neuronal artificial es la más óptima para tareas de reconocimiento de emociones en la voz, diferentes estudios expresan opiniones variadas. Para este proyecto se decide empezar por aplicar redes de tipo CNN, con el propósito de explotar sus capacidades de extracción de características propias y eliminación de ruido a través de la aplicación de filtros y MaxPooling. Las CNN reducen los datos de voz a sus características clave y utilizan las probabilidades combinadas de las características identificadas que aparecen juntas para determinar una clasificación.

Tras la fase de pruebas con diferentes modelos con estructuras puramente CNN, el modelo final del presente proyecto está constituido por una solución hibrida CNN y el LSTM (Long-Short Term Memory) que combina los beneficios de una estructura *feed-forward* junto con una retroalimentaria. Esta arquitectura 1D CNN-LSTM se resume en la figura 22.

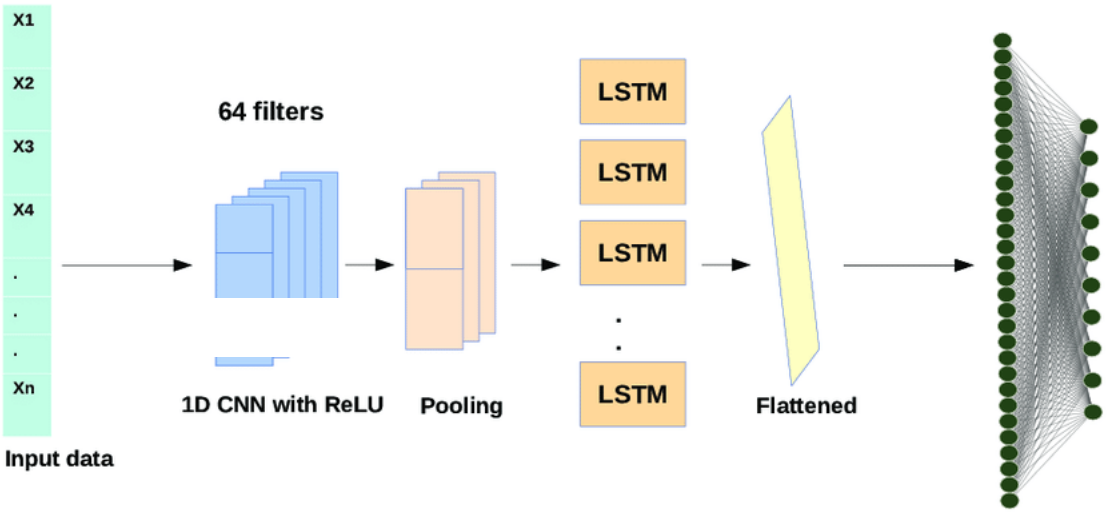


Figura 22 Arquitectura de un modelo híbrido 1D CNN + LSTM (Hamad, R. A., Yang, L., Woo, W. L., & Wei, B., 2020)

Tras probar diferentes opciones, detalladas en el apartado de resultados, se pudieron comprobar los beneficios de esta combinación entre ambas redes. En contraste con los modelos actuales que asumen un campo receptivo espaciotemporal fijo o un promedio temporal simple para el procesamiento secuencial, los modelos convolucionales recurrentes son "doblemente profundos" en el sentido de que pueden estar compuestos por "capas" espaciales y temporales. Estos modelos pueden tener ventajas cuando los conceptos son complejos y / o los datos de entrenamiento son limitados, ambos factores del presente trabajo. Esta arquitectura define dos submodelos: el modelo CNN para la extracción de características y el modelo LSTM para interpretar las características a lo largo del tiempo [70].

Para el desarrollo de las redes neuronales artificiales se trabaja estrechamente con la librería Keras, una de las librerías más populares de Redes Neuronales en Python. Es capaz de correr sobre Tensorflow, otra librería destacada de Machine Learning. Keras es ideal para experimentación con Deep Learning [71].

Se probaron diferentes modelos probando entre estructuras CNN, CNN-LSTM y experimentando con los optimizadores y los *callbacks*, además de probar extrayendo diferentes características.

El modelo final consiste en una arquitectura 1D CNN-LSTM con los parámetros plasmados en la tabla 7.

Nº epochs	**75**
Batch size	48
Optimizer	Adam

loss	Categorical crossentropy
Metrics	Accuracy
Callbacks	ReduceLROnPlateau
Mínimum learning rate	0.000001
Monitor metric	loss

Tabla 7 Hiperparámetros del modelo final 1D CNN + LSTM

Este modelo se aplica a un conjunto de datos con 50 coeficientes MFCC extraídos de cada audio, para detectar una de las ocho emociones ofrecidas por los datasets utilizados.

El método de aprendizaje transversal implementado da lugar a un modelo de 13 capas, descritas en la tabla 8.

Nº Capa	**Nombre**	**Descripción**
1	Conv1D	Filter size= 256 Kernel size= 6 Strides = 1 Padding = same Activation = relu Al ser la primera capa, se establecen las dimensiones del audio de entrada
2	AveragePooling1D	Pool size = 4 Strides = 2 Padding = same
3	Conv1D	Filter size= 128 Kernel size= 6 Strides = 1 Padding = same Activation = relu

4	AveragePooling1D	Pool size = 4 Strides = 2 Padding = same
5	Conv1D	Filter size= 128 Kernel size= 6 Strides = 1 Padding = same Activation = relu
6	AveragePooling1D	Pool size = 4 Strides = 2 Padding = same
7	Dropout	20% droput
8	LSTM	128 hidden units Activation = relu
9	Dropout	20% dropout
10	Flatten	
11	Dense	Units = 32 Activation = relu
12	Dropout	30% dropout
13	Dense	Units = 8 Activation = softmax

Tabla 8 Arquitectura seleccionada para la red neuronal artificial detrás del sistema final

El modelo CNN-LSTM lo componen cuatro tipos de capas importantes:

- **Conv1D**: al estar trabajando con señales de audio se emplean filtros convolucionales de 1 dimensión. Esta capa identifica regiones destacadas entre intervalos y activa ciertas características del audio, generando el mapa de características.
- **AveragePooling1D**: se utiliza como técnica de *downsampling*; simplifica la salida disminuyendo la tasa de muestreo tomando el valor promedio sobre la ventana definida

por pool size. Esto reduce el número total de parámetros que la red necesita para aprender. Esta capa ayuda a evitar el problema de *overfitting*.

- **Unidad lineal rectificada (ReLU)**: da lugar a un entrenamiento más rápido y eficaz induciendo no linealidad.
- **LSTM**: aprende las dependencias a largo plazo en series de tiempo y datos secuenciales. Esto añade un componte que puede "recordar" estados previos y utilizar esta información para decidir cuál será el siguiente.
- **Dropout**: técnica de regularización que reduce la complejidad del modelo eliminando unidades tanto ocultas como visibles antes de pasar a la siguiente capa. Esto ayuda al modelo a generalizar y también reduce el *overfitting*.
- **Flatten:** se emplea antes de las fully-connected layers del final para aplanar los datos y reducir sus dimensiones para poder ser recogidos por las capas Dense.
- **Dense**: esta es la capa que realiza la clasificación. Todos los nodos están conectados con todos las de la capa anterior.
- **Softmax**: devuelve la probabilidad de que cada una de las clases sea cierta, y es la función de activación adecuada para problemas de multi clasificación.

La figura 23 muestra la dimensión de los datos a medida que transcurren a través de la arquitectura.

Ya que los datos están distribuidos de manera equilibrada, una métrica adecuada para comparar la destreza de los distintos modelos es la eficacia (*model accuracy*). Un análisis de los resultados obtenidos y la comparativa entre modelos se detalla en la sección 4.6 Resultados del Proyecto.

La figura 24 muestra la estructura del código que compone el proceso de entrenamiento y sus fases previas. Todo el código que compone la parte de Inteligencia Artificial de este proyecto se desarrolla en Jupyter Notebook.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 58, 256)           1792
_________________________________________________________________
average_pooling1d (AveragePo (None, 29, 256)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 29, 128)           196736
_________________________________________________________________
average_pooling1d_1 (Average (None, 15, 128)           0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 15, 128)           98432
_________________________________________________________________
average_pooling1d_2 (Average (None, 8, 128)            0
_________________________________________________________________
dropout (Dropout)            (None, 8, 128)            0
_________________________________________________________________
lstm (LSTM)                  (None, 8, 128)            131584
_________________________________________________________________
activation (Activation)      (None, 8, 128)            0
_________________________________________________________________
dropout_1 (Dropout)          (None, 8, 128)            0
_________________________________________________________________
flatten (Flatten)            (None, 1024)              0
_________________________________________________________________
dense (Dense)                (None, 32)                32800
_________________________________________________________________
dropout_2 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 264
=================================================================
Total params: 461,608
Trainable params: 461,608
Non-trainable params: 0
```

Figura 23 Arquitectura de la estructura seleccionada y las dimensiones de los datos de salida de cada capa

/Anaconda Jupyter Notebook con Lenguaje Python

// 1- Ingesta y preprocesado de datos

1. Ingesta de audio
2. Extracción del espectrograma y la forma de onda
3. Creación de datos sintéticos (Data Augmentation)
4. Extracción de características MFCC utilizando la librería *librosa* (Feature Extraction)

// 2- Entrenamiento

5. Mezcla aleatoria de los datos, división entre entrenamiento y test
6. Preparado de datos para ingesta del algoritmo y PCA
7. Estructura de la red neuronal artificial
8. Entrenamiento del algoritmo con los datos
9. Serializar el modelo a JSON y los *weights* a HDF5
9. Predicción de la emoción humana a partir de los datos de entrenamiento

Figura 24 Esquema del script desarrollado para el modelado del algoritmo de reconocimiento de emociones

A continuación, se muestra una tabla con las librerías empleadas durante todo el proceso de construcción del modelo.

Librería	**Descripción**
os	Librería de Python que permite interactuar con el sistema operativo
pandas	Librería de Python especializada en el manejo y análisis de estructuras de datos
matplotlib	Librería de visualización de datos y trazado gráfico para Python
numpy	Librería de Python para trabajar con arrays y en el dominio del algebra lineal
seaborn	Librería de Python para visualización de datos basada en matplotlib. Especializada en gráficas y representaciones estadísticas
librosa	Librería de Python para el análisis de audio y música. Incluye capacidades de extracción de datos/características de los archivos de audio.
sklearn	Librería open source para aprendizaje automático en Python
Imblearn	Librería open source basada en sklearn que provee herramientas para tareas de clasificación con datos desbalanceados
tensorflow	Librería open source para aprendizaje automático en Python
keras	Librería open source de redes neuronales escrita en Python

Tabla 9 Listado y descripción de las librerías empleadas en Python durante el proyecto

3.5.6 Desarrollo web

Una vez se ha completado el entrenamiento y se obtiene un modelo final, se plantea un sistema que pueda recoger datos de entrada y recibir las predicciones del modelo sobre esos datos de entrada, con capacidad de respuesta en tiempo real e integrado con un componente front-end. La arquitectura propuesta se muestra en la figura 25.

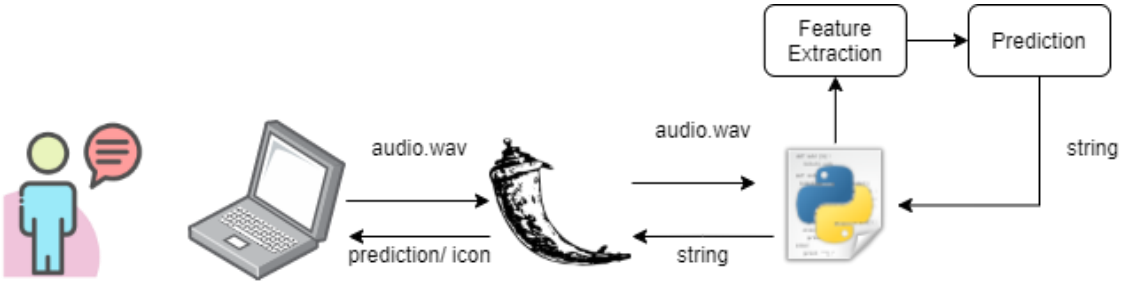


Figura 25 Estructura y flujo de datos de la aplicación web

Para poder servir las predicciones a través de una aplicación práctica, se escoge Flask como *framework* para realizar un desarrollo web. Flask facilita el desarrollo de aplicaciones web en Python, el lenguaje principal utilizado en este proyecto.

El objetivo es montar una dinámica cliente servidor, durante la cual una solicitud, en este caso una grabación de audio se envía de cliente a servidor en forma de URL como HTTP POST. Después de eso, se recibe una respuesta del servidor en forma de recurso HTML e imágenes. Flask facilita la configuración de funciones de Python que se pueden invocar a través de la web con estas solicitudes.

Inicialización del proyecto

La instalación (a través del gestor de paquetes **pip**) y configuración del servidor es una parte primordial para preparar un entorno de trabajo que satisfaga las necesidades computacionales de la aplicación. Sin embargo, esto es de utilidad a la hora de hacer un despliegue en servicios de nube como Heroku o GCP para que estos sepan que dependencias son necesarias para lanzar la aplicación.

1. Instalar y activar adecuadamente *virtualenv* para poder tener varios entornos Python en paralelo. De esta manera, se evitan problemas de compatibilidad entre las diferentes versiones de las bibliotecas. Por ejemplo, Tensorflow solo funciona bajo Python 3.7.
2. Instalar *Flask* con el comando **pip**.
3. Instalar los paquetes especificados en el fichero *requirements.txt*
4. Inicializar el repositorio Git

El proyecto se organiza siguiendo la jerarquía dentro del directorio *webapp* mostrada a continuación:

```
+ webapp
|_ + static
      |_ + js
            |_ app.js
      |_ + styles
            |_ style.css
      |_ icons.png
      |_ logo.png

|_ + templates
      |_ index.html
      |_ prediction.html

|_ app.py
|_ audio.wav
|_ models
|_ requirements.txt
|_ imagen.jpg
```

Figura 26 Jerarquía del directorio donde se despliega la aplicación

Diseño y creación de Marca

Nombre de Marca: **EMoody**

Las páginas de predicciones (prediction.html) están diseñadas para que cada emoción muestre un fondo de pantalla diferente, acotándose a la estrella de emociones de Plutchik (1980) [15]. A su vez, se devuelve un icono característico de dicha emoción, así se consigue ofrecer una experiencia más divertida y visual.

Se diseña un logo personalizado para la marca EMoody (figura 27). Para crearlo se utiliza una plantilla y se personaliza utilizando Adobe Illustrator (2020).

Todos los recursos visuales (fondos de pantalla, iconos y plantillas) se descargan de freepik bajo la licencia gratuita de uso personal.

Figura 27 Logo EMoody

Tecnologías empleadas

Lenguajes de programación:

- Python
- HTML
- CSS
- JavaScript

IDs

- Jupyter Notebook
- Visual Studio Code

Distribuciones

- Anaconda

Frameworks

- Flask

Control de versiones

- Git

Gestor de paquetes

- Pip

Diseño

- Illustrator

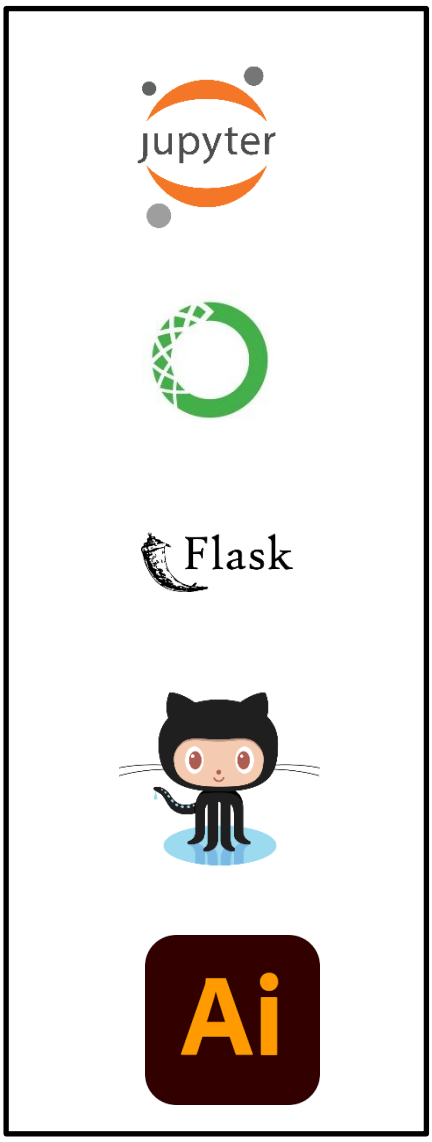


Figura 28 Logos de las tecnologías empleadas

Aplicación

// app.py: contiene el código de la aplicación (aquí se crea la app y sus vistas)

1. Crear objeto Flask que identifique la app y que asocie las vistas a las rutas

2. Importar funciones necesarias para le extracción de características (los mismos 50 coeficientes MFCC), carga de modelos y predicciones

3. Crear rutas de la aplicación:

- @app.route("/) : ruta base- guarda y nombra el archivo de audio

- @app.route("/predictMixed"): llamada al algoritmo mixto, clasificación de la emoción, redirecciona a página de resultados tras predicción con el modelo mixto

- @app.route("/predictFemale"): llamada al algoritmo femenino, clasificación de la emoción, redirecciona a página de resultados tras predicción con el modelo femenino

- @app.route("/predictMale"): llamada al algoritmo masculino, clasificación de la emoción, redirecciona a página de resultados tras predicción con el modelo masculino

4. Establecer host, puerto y modo debug

Figura 29 Esquema del script app.py (ejecución de la web app)

@app.route se encarga de convertir una función Python regular en una *función vista* de Flask como respuesta HTTP mostrada mediante un cliente HTTP (navegador web). Pasando los valores *'/', '/predictMixed', '/predictFemale'* y *'/predictMale'* a **@app.route()** se le indica que esta función debe responder a las solicitudes web para esa URL.

La aplicación se lanza haciendo uso del comando **$ flask run** en el directorio de trabajo. De forma predeterminada, Flask ejecuta la aplicación en el puerto 5000. La aplicación se encuentra en la ruta http://localhost:5000 del navegador (figura 64 de la sección de resultados).

Plantillas HTML

Flask proporciona la función **render_template()** que permite hacer uso del motor de plantillas Jinja. Con esta funcionalidad se escribe HTML en archivos .html separados de manera cómoda, sencilla y más organizada. Estas *plantillas* se llaman en el **return** al final de **@app.route** para crear y mostrar todas las páginas de la aplicación como resultado de los *requests*.

En este proyecto se desarrollan 2 plantillas:

1. **index.html**: página principal (figuras 64 y 65 de la sección resultados)

2. **prediction.html**: página de resultados de las predicciones

Se utiliza el kit de herramientas Bootstrap por su destreza en el desarrollo de web responsive con HTML, CSS y JavaScript. Para instalar y utilizar Bootstrap el sitio web se hace uso de uno de los enlaces del CDN (*Content Delivery Network*) facilitados en su web oficial. De esta manera los archivos del framework se cargan sin tenerlos alojados en el servidor. La arquitectura de la parte orientada hacia el cliente se representa en la figura 30 a continuación.

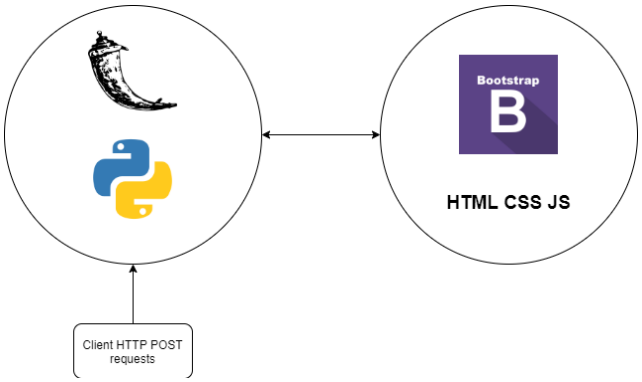


Figura 30 Arquitectura que sostiene a la interfaz de usuario

Recorder.js

Para realizar la tarea de grabación de audio se utiliza la biblioteca Recorder.js de JavaScript. Esta librería permite guardar los audios en contenedores *.wav*, el formato de ingesta de los modelos desplegados. Aquí se usa junto con **getUserMedia()** para poder grabar audio desde el micrófono de un usuario (u otros dispositivos de entrada) directamente en el sitio web. Para integrar un sistema de grabación de audio en tiempo real se tienen en cuenta los siguientes archivos:

1. **index.html**: aquí se encuentra la interfaz de usuario

2. app.js: código de la aplicación de grabación. Se divide en 4 funciones importantes:

- **startRecording():** lanza **getUserMedia()** y convierte el audio en un AudioContext que se pasa al objeto **Recorder.js.**
- **stopRecording():** paraliza el proceso de grabación y el acceso al micrófono, desencadena el proceso de exportación a *.wav* y habilita el botón de grabación de nuevo.
- **pauseRecording():** determina si la grabación está en proceso y permite pausar y resumir
- **createDownloadLink(blob):** recibe el audio en formato blob. Permite escuchar la grabación en el navegador, descargarlo y subirlo al servidor (en este caso se hace un POST a "/" donde se guarda el fichero en el directorio para ser leído por los modelos). Para ello se utiliza **XMLHttpRequest** ya que es compatible con todos los navegadores compatibles getUserMedia(). El código crea un enlace de carga que, al hacer clic, publicará el blob y el nombre del archivo en el script *app.py* del lado del servidor.

3. recorder.js: se carga en el *index.html* a través de la URL de producción de Rawgit *cdn.rawgit.com*

3.6 Recursos requeridos

Los recursos requeridos para este proyecto son únicamente computacionales. Estos se dividen en componentes hardware y software:

- Ordenador MSI (Windows) para desplegar la arquitectura. Aquí se instalan todo el software para poder realizar todas las partes del proyecto:
    - Anaconda con *Jupyter Notebook* para guardar los datos de entrenamiento y entorno de programación para el preprocesado de datos, entrenamiento de los modelos y predicciones.
    - Se emplea *anaconda prompt* para descargar todas las librerías necesarias, usando el comando **pip**. Debido a la naturaleza más compleja de las dependencias de *tensoflow* y *keras*, se crea un *environment* especial donde se desarrollan los modelos de aprendizaje *Deep Learning* con estas librerías.
    - La aplicación web se desarrolla en Visual Studio Code. Aquí se trabaja el framework Flask. Para lanzar la aplicación también se utiliza *anaconda prompt* ya que se necesita el mismo *environment* de trabajo utilizado para el desarrollo del algoritmo.
- Portátil con micrófono: para poder llevar a cabo la grabación de voz e interactuar con la aplicación web.

3.7 Presupuesto

Tipo de coste	**Valor**	**Comentarios**
Horas de trabajo en el proyecto	470 horas	Duración aproximada del proyecto de 4 meses
Equipo técnico utilizado		
Ordenador portátil- MSI Prestige 14 A10RB-020ES	1349€	Recurso ya adquirido. Precio actual en el mercado si se tuviese que adquirir nuevo.
Monitor- Asus 23.8 pulgadas	129,99€	Recurso ya adquirido. Precio actual en el mercado si se tuviese que adquirir nuevo.
Software utilizado		

Jupyter notebook (Ipython Notebook)	0€	Software libre
Adobe Illustrator	0€	Se ha trabajado bajo la prueba gratuita de 7 días. Una vez esta expira son 24,19 € al mes
Visual Studio	0€	Software libre
Frameworks y librerías	0€	Software libre
Fuentes de investigación		
Estudios e informes	0€	Todos los informes y estudios de investigación empleados para realizar este proyecto eran gratuitos.

Tabla 10 Estimación del presupuesto del trabajo realizado

3.8 Viabilidad

El estado actual del proyecto consiste en un prototipo del producto que se pondría en producción en un futuro. Aun no se puede considerar un producto monetizable, pero en el apartado de Anexos se ha incluido un *business model canvas* con una propuesta inicial del posible modelo de negocio que se llevaría a cabo con la aplicación desarrollada.

3.9 Resultados del proyecto

3.9.1 Ingesta, limpieza y análisis exploratorio de datos

Una parte primordial del proceso de verificación de los datos para el entrenamiento es comprobar que haya equilibrio entre las clases. Para ello se hace un conteo de ellas utilizando elementos visuales mostrados en la figura 31. Se hace un conteo tanto para las emociones de los hombres como de las mujeres, y se descubre que ambas se comportan igual; se identifica un destacado desbalance de clases que puede obstaculizar el proceso de entrenamiento.

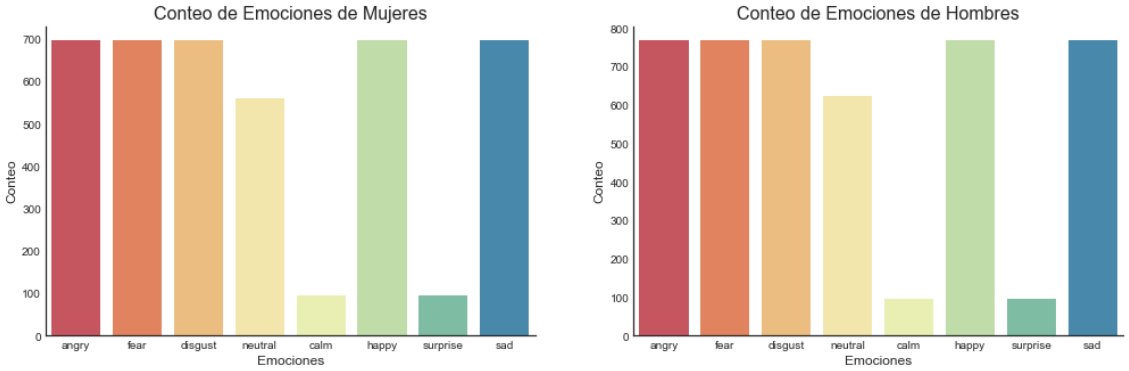


Figura 31 Diagrama de barras mostrado el conteo de cada emoción antes de aplicar oversampling a los datasets femenino (izquierda) y masculino (derecha)

Existe una carencia de emociones de *calm* y de *surprise*. Es un desafío para las tareas de modelado predictivo trabajar con una distribución de clases severamente sesgada, causando desigualdad en los costos de la clasificación errónea [72].

Tras haber realizado el preprocesamiento al completo de los datos, incluyendo las técnicas de *oversampling*, generación de datos sintéticos y extracción de características, el dataset ya está preparado para poder pasar a la fase de entrenamiento. En este punto se trabaja con un dataset que contiene la misma cantidad de datos (figura 32) para cada emoción, integrado con nuevos datos sintéticos, y al conjunto entero se le han extraído las características que van a permitir al modelo a continuación entender los datos.

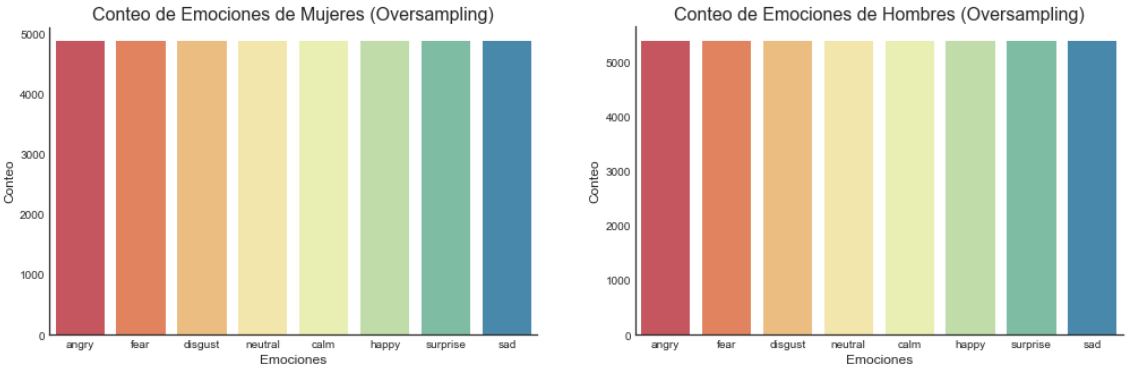


Figura 32 Diagrama de barras mostrado el conteo de cada emoción después de aplicar oversampling a los datasets femenino (izquierda) y masculino (derecha)

Gráficos de onda y espectrogramas

Para visualizar los archivos con los que se trabaja durante el desarrollo del modelo, se emplean los siguientes gráficos. Es interesante realizar una comparativa de cómo se comporta una misma emoción en la voz de un hombre comparado con una mujer. Las figuras a continuación muestran las emociones *fear* y *happy* para ambos sexos. En efecto, tienen formas de onda diferentes y los

espectrogramas muestran sutiles diferencias en el reparto de energías de sus contenidos frecuenciales.

Figura 33 Gráfico de onda de una mujer expresando la emoción miedo

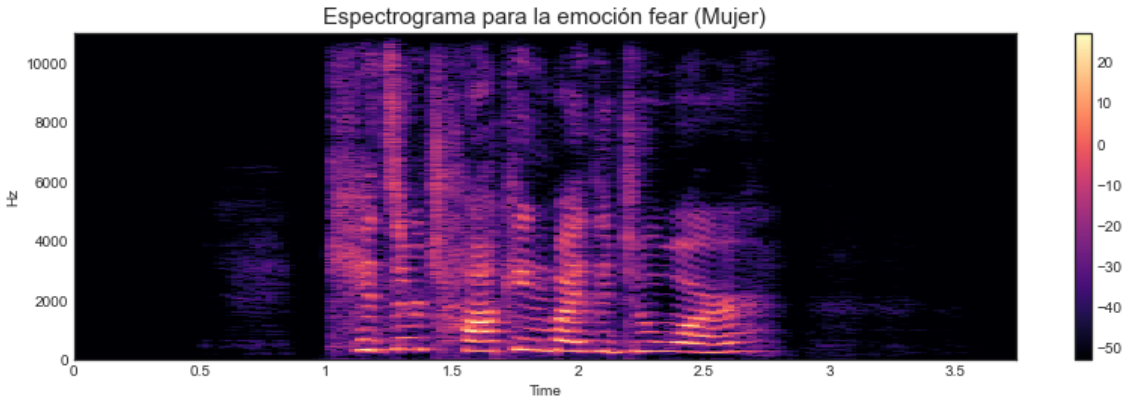


Figura 34 Espectrograma para la emoción de miedo (mujer)

Figura 35 Gráfico de onda de un hombre expresando la emoción miedo

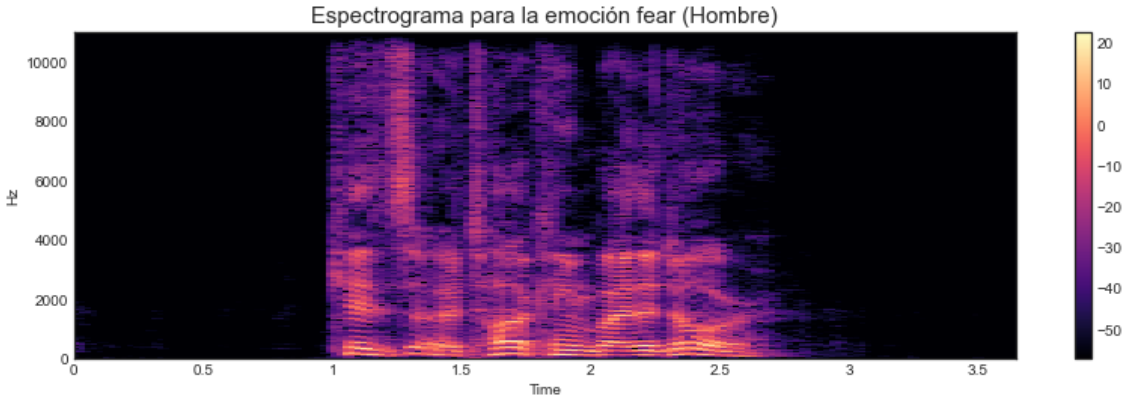


Figura 36 Espectrograma para la emoción de miedo (hombre)

Figura 37 Gráfico de onda de una mujer expresando la emoción felicidad

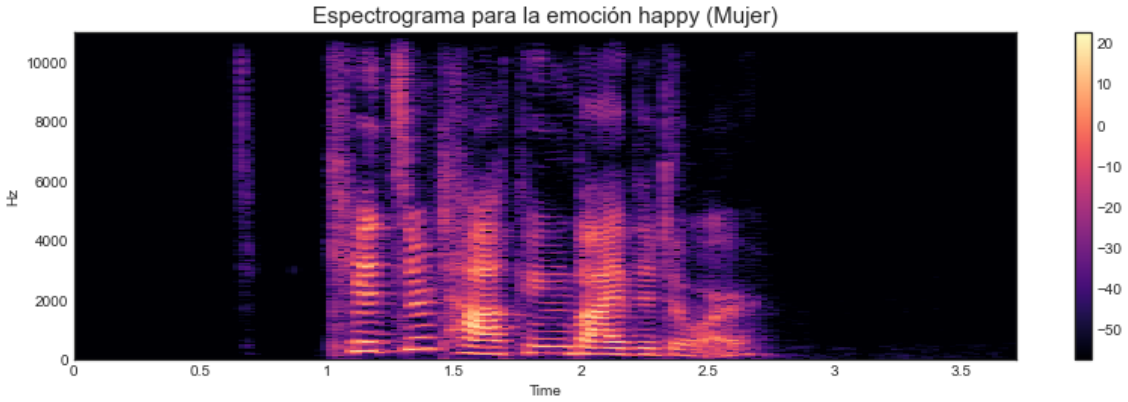


Figura 38 Espectrograma para la emoción de felicidad (mujer)

Figura 39 Gráfico de onda de un hombre expresando la emoción felicidad

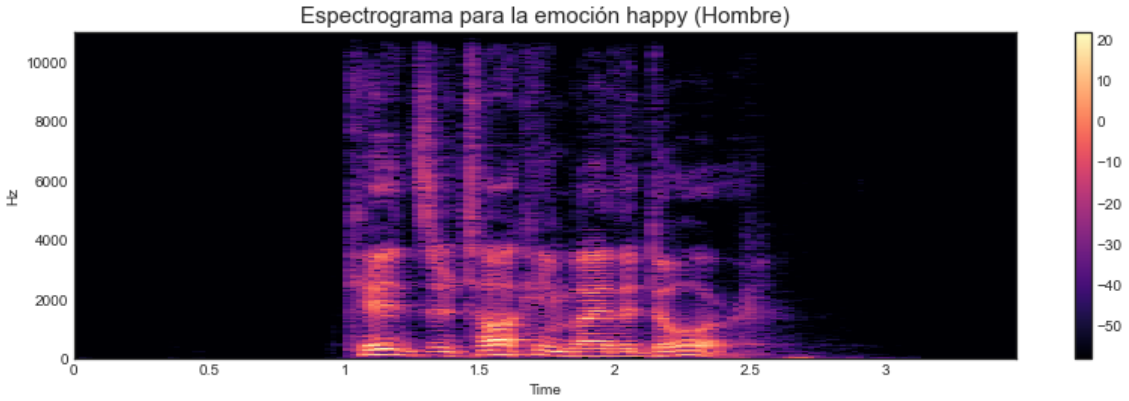


Figura 40 Espectrograma para la emoción de felicidad (hombre)

Gráficos de onda tras aplicar Data Augmentation

Tras la creación de nuevas tandas de datos sintéticos con diferentes modificaciones, es interesante observar el impacto que estas han tenido sobre la onda inicial (figura 41). Todas las gráficas de ondas a continuación son sobre la grabación de una mujer con emoción neutral.

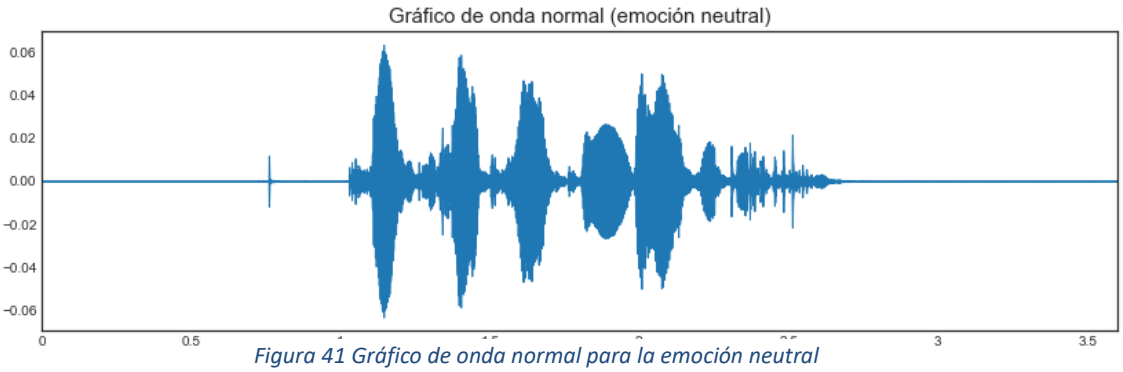


Figura 41 Gráfico de onda normal para la emoción neutral

Figura 42 Gráfico de onda de la emoción neutral con ruido

Figura 44 Gráfico de onda de la emoción neutral con onda alargada

Figura 43 Gráfico de onda de la emoción neutral con onda desplazada

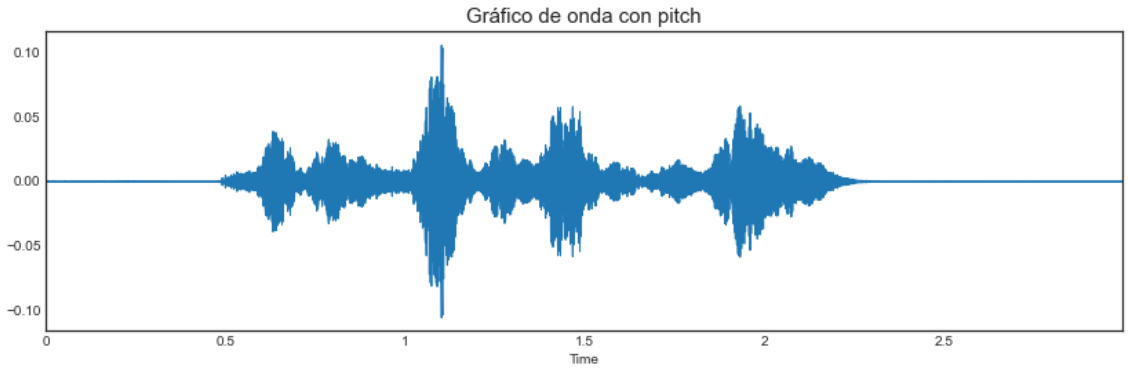


Figura 47 Gráfico de onda de la emoción neutral con cambio de tono

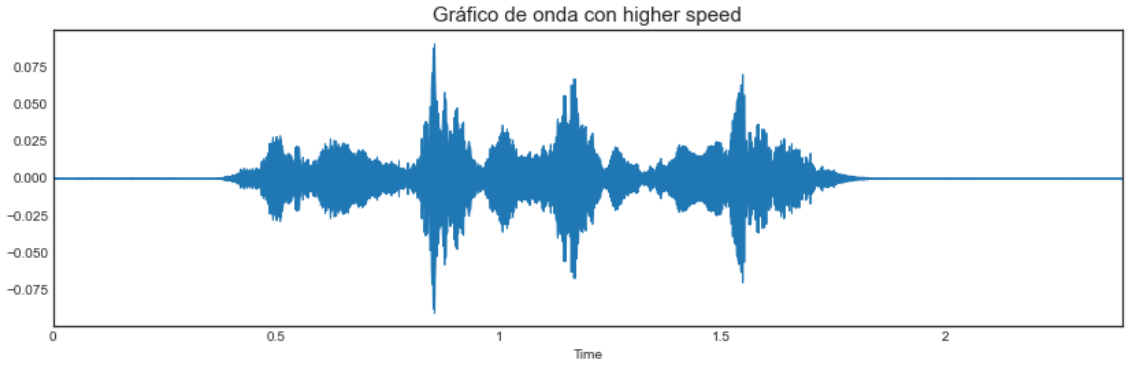


Figura 46 Gráfico de onda de la emoción neutral con velocidad aumentada

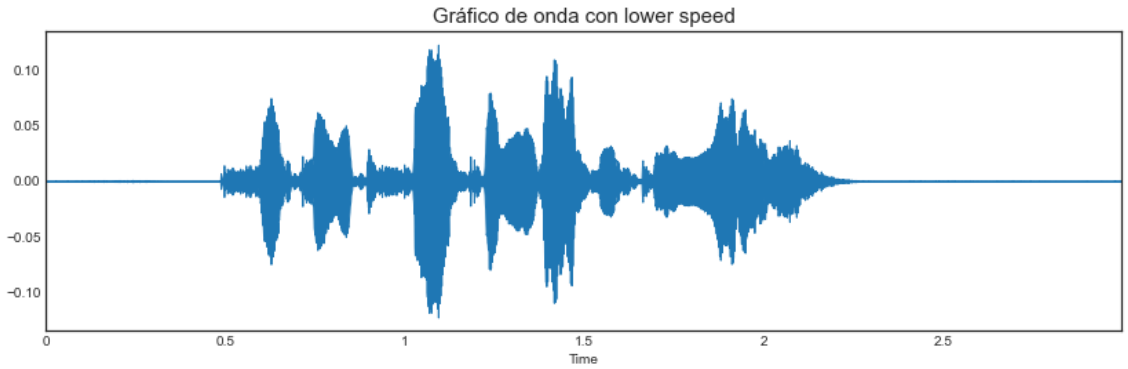


Figura 45 Gráfico de onda de la emoción neutral con velocidad reducida

La inyección de ruido es la acción que menos distorsiona la onda, y es curioso como con el resto de las operaciones se obtiene una forma de onda muy similar. Esto puede resultar ser beneficioso para que el modelo no se confunda en diferentes situaciones ambientales.

3.9.2 Extracción y selección de características

Feature Extraction: MFCC

Para determinar el número de MFCC que seleccionar de los datos, se hace un análisis previo de los espectrogramas tras haber aplicado la extracción de características MFCC. La figura 48 muestra el resultado tras extraer 20 coeficientes y la figura 49 con 50.

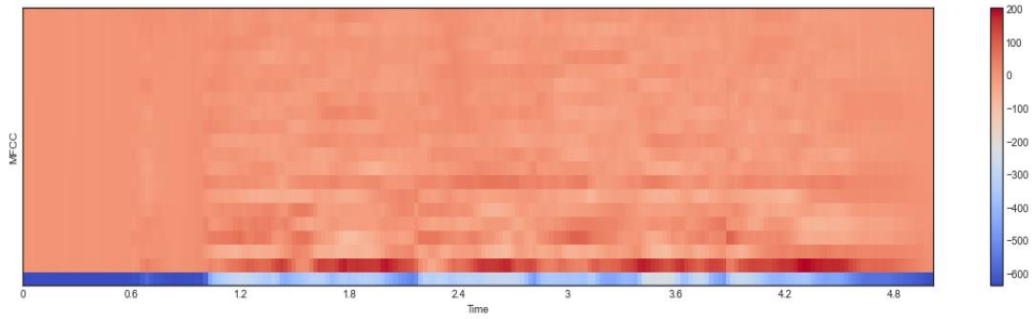


Figura 48 Espectograma de MFCCs de un audio de muestra: 03-01-05-02-01-01-08.wav, emoción de enfado de una mujer. Extracción de 20 coeficientes

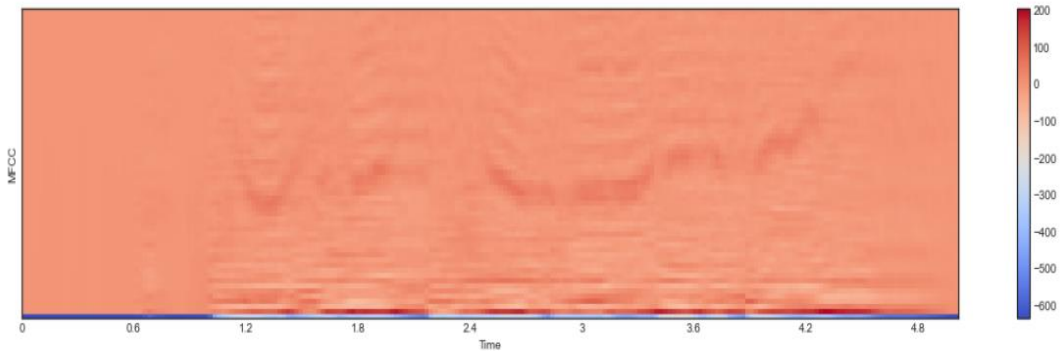


Figura 49 Espectrograma de MFCCs de un audio de muestra: 03-01-05-02-01-01-08.wav, emoción de enfado de una mujer. Extracción de 50 coeficientes

Extrayendo 50 coeficientes (figura 49) se distinguen más formantes. Las zonas oscuras son las que muestran formantes en el espectro; los sonidos se pueden identificar mucho mejor a través de los formantes y sus transiciones. Por lo tanto, se opta por la extracción de 50 MFCCs de cada muestra de audio.

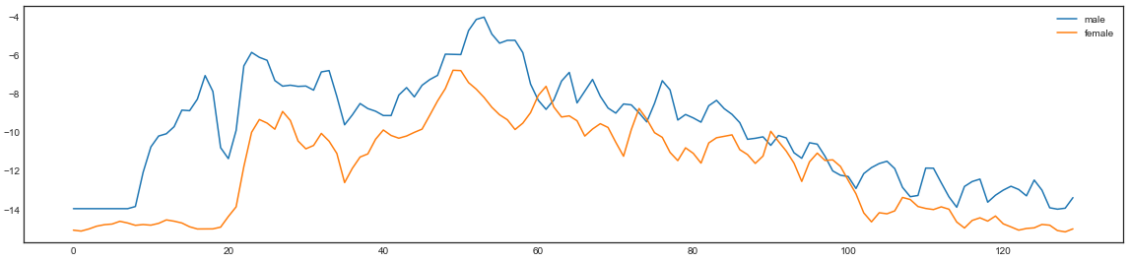


Figura 48 Gráfico de líneas con los coeficientes MFCC para hombres y mujeres a lo largo del mismo periodo de tiempo

A su vez, se hace una comparativa directa entre el comportamiento de los MFCC extraídos para ambos sexos. Para ello se genera un gráfico de líneas (figura 50) con la media de los MFCC para cada instante del tiempo como una serie temporal. Se distinguen claras diferencias en el patrón de hombres y mujeres, siendo ellos quienes exhiben un tono más elevado. Dadas las diferencias encontradas entre las voces de ambos sexos, se considera que su división de cara al entrenamiento puede ser de utilidad para el algoritmo a la hora de distinguirlos y por lo tanto clasificarlos mejor.

Feature Selection: PCA

Para comprobar si se es necesario reducir la dimensión del dataset de entrenamiento, se aplica el análisis de PCA de varianza explicada. Se utilizan gráficos de varianza acumulada a lo largo de las características extraídas para los datos de entrenamiento.

Sin separación por sexos, extracción de ZCR, Chroma Shift, MFCC, RMSV, Mel Spectogram

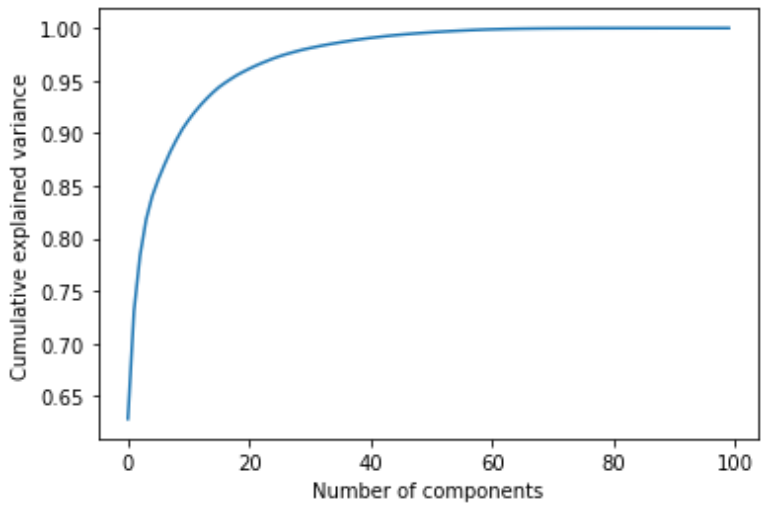


Figura 49 PCA: Sin separación por sexos, extracción de ZCR, Chroma Shift, MFCC, RMSV, Mel

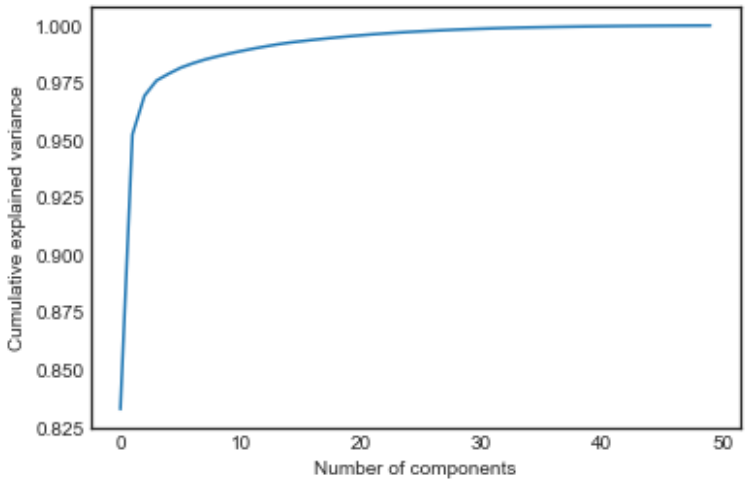


Figura 50 PCA: Separación por sexo, 50 MFCCs

Esta curva muestra que aproximadamente los primeros 5 componentes contienen aproximadamente el 95% de la varianza, mientras que necesitan alrededor de 40 componentes para describir cerca del 100% de la varianza. Esta gráfica muestra que puede existir cierta redundancia, pero los resultados de los modelos más adelante demuestran que 50 coeficientes inyectan destreza a los modelos. Al tratarse de datos de audio puede existir una variabilidad muy alta, por lo que extraer muchos coeficientes puede ayudar a distinguir sutilezas en los datos. Prueba de ello son los resultados obtenidos con los datasets de 13 y 30 coeficientes extraídos respectivamente (tablas 11, 12, 13).

3.9.3 Modelo y Entrenamiento

Se probaron diferentes estructuras de redes neuronales, combinando diferentes técnicas de extracción de características, tipos de redes neuronales artificiales, y combinación de estas. Los detalles de esta experimentación, incluyendo los resultados de cada prueba, se resumen en las tablas a continuación.

Features extraídos	Arquitectura	Optimizador	Validation Acc.	Validation Loss
ZCR, Chroma Shift, MFCC, RMSV, Mel Spectogram	2 layers Conv1D + MaxPooling + Dense + Flatten	Adam	0.68	1.03
ZCR, Chroma Shift, MFCC, RMSV, Mel Spectogram	4 layers Conv1D + MaxPooling + Flatten	Adam	0.71	1.32

ZCR, Chroma Shift, MFCC, RMSV, Mel Spectogram	2 layers Conv1D+ BatchNormalisation + 1 layer LSTM + MaxPooling + Dense + Flatten	Adam	0.72	0.97
ZCR, Chroma Shift, MFCC, RMSV, Mel Spectogram	2 layers Conv1D + BatchNormalisation + 1 layer LSTM + MaxPooling + Dense + Flatten	Adam + ReduceLROnPlateau	0.77	0.82
MFCC	4 layers Conv1D + 1 layer LSTM + AvgPooling + Flatten + 2 layers Dense	Adam + ReduceLROnPlateau	0.87	0.51
MFCC (50)	**3 layers Conv1D** +1 layer LSTM + **AvgPooling + Flatten + 2 layers Dense**	**Adam + ReduceLROnPlateau**	**0.90**	**0.37**
MFCC (30)	3 layers Conv1D + AvgPooling + Flatten + 2 layers Dense	Adam + ReduceLROnPlateau	0.89	0.36
MFCC (13)	3 layers Conv1D + AvgPooling + Flatten + 2 layers Dense	Adam + ReduceLROnPlateau	0.79	0.5

Tabla 11 Arquitectura de los modelos entrenados para el conjunto de datos de género mixto y los resultados de validation accuracy y los para cada etapa

Features extraídos	**Arquitectura**	**Optimizador**	**Validation Acc.**	**Validation Loss**
MFCC (50)	4 layers Conv1D + AvgPooling + Flatten + 2 layers Dense	Adam + ReduceLROnPlateau	0.92	0.36
MFCC (50)	**3 layers Conv1D + AvgPooling + Flatten + 2 layers Dense**	**Adam + ReduceLROnPlateau**	**0.93**	**0.35**
MFCC (30)	3 layers Conv1D + AvgPooling + Flatten + 2 layers Dense	Adam + ReduceLROnPlateau	0.90	0.36
MFCC (13)	3 layers Conv1D + AvgPooling + Flatten + 2 layers Dense	Adam + ReduceLROnPlateau	0.85	0.44

Tabla 12 Arquitectura de los dos últimos modelos entrenados para el conjunto de datos de género femenino y los resultados de validation accuray y los para cada etapa

Features extraídos	Arquitectura	Optimizador	Validation Acc.	Validation Loss
MFCC	4 layers Conv1D + AvgPooling + Flatten + 2 layers Dense	Adam + ReduceLROnPlateau	0.91	0.51
MFCC	**3 layers Conv1D + AvgPooling + Flatten + 2 layers Dense**	**Adam + ReduceLROnPlateau**	**0.91**	**0.4**
MFCC (30)	3 layers Conv1D + AvgPooling + Flatten + 2 layers Dense	Adam + ReduceLROnPlateau	0.87	0.52
MFCC (13)	3 layers Conv1D + AvgPooling + Flatten + 2 layers Dense	Adam + ReduceLROnPlateau	0.8	0.52

Tabla 13 Arquitectura de los modelos entrenados para el conjunto de datos de género masculino y los resultados de Validation Accuracy y Loss para cada etapa

A su vez se visualizan las gráficas de las métricas de Accuracy y Loss extraídas durante las fases de entrenamiento y de validación. La figura 53 a continuación muestra los resultados obtenidos para estas métricas durante el entrenamiento de la entrada (2) de la tabla 911

- **Features extraídos**: ZCR, Chroma Shift, MFCC, RMSV, Mel Spectogram
- **Arquitectura**: 4 layers Conv1D + MaxPooling + Flatten
- **Optimizador**: Adam
- **Validation Accuracy**: 0.71
- **Validation Loss**: 1.32

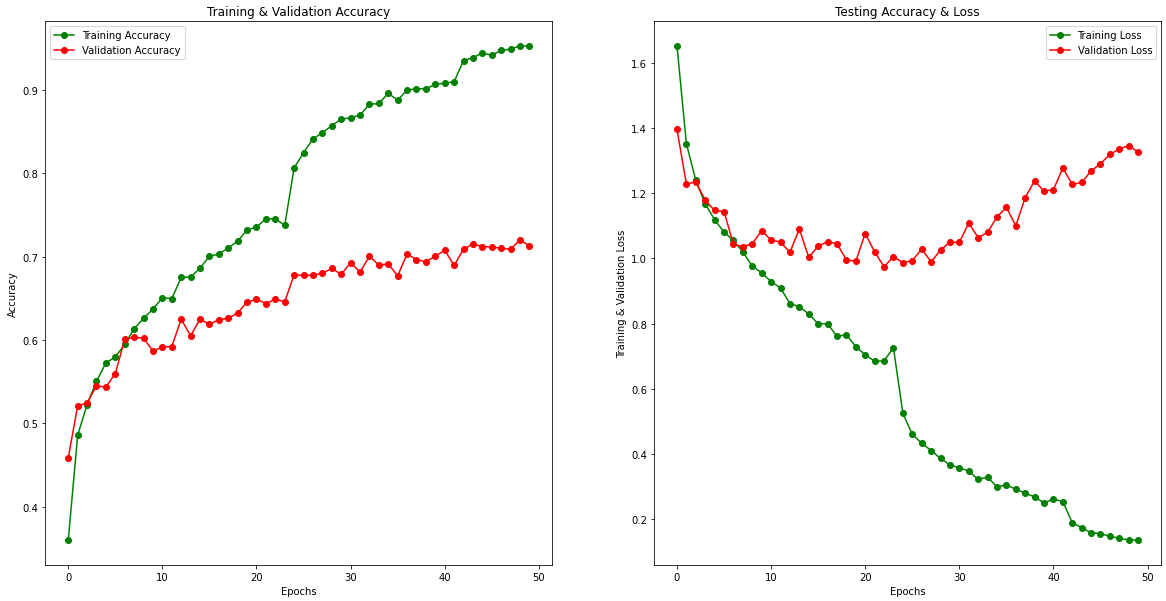


Figura 51 Gráficas de Accuracy y Loss durante el entrenamiento del segundo modelo de la tabla 11.

La figura 54 muestra los resultados de entrenamiento de la entrada (4) de la tabla 11:

- **Features extraídos**: ZCR, Chroma Shift, MFCC, RMSV, Mel Spectogram
- **Arquitectura**: 2 layers Conv1D + BatchNormalisation + 1 layer LSTM + MaxPooling + Dense + Flatten
- **Optimizador**: Adam
- **Validation Accuracy**: 0.77
- **Validation Loss**: 10.82

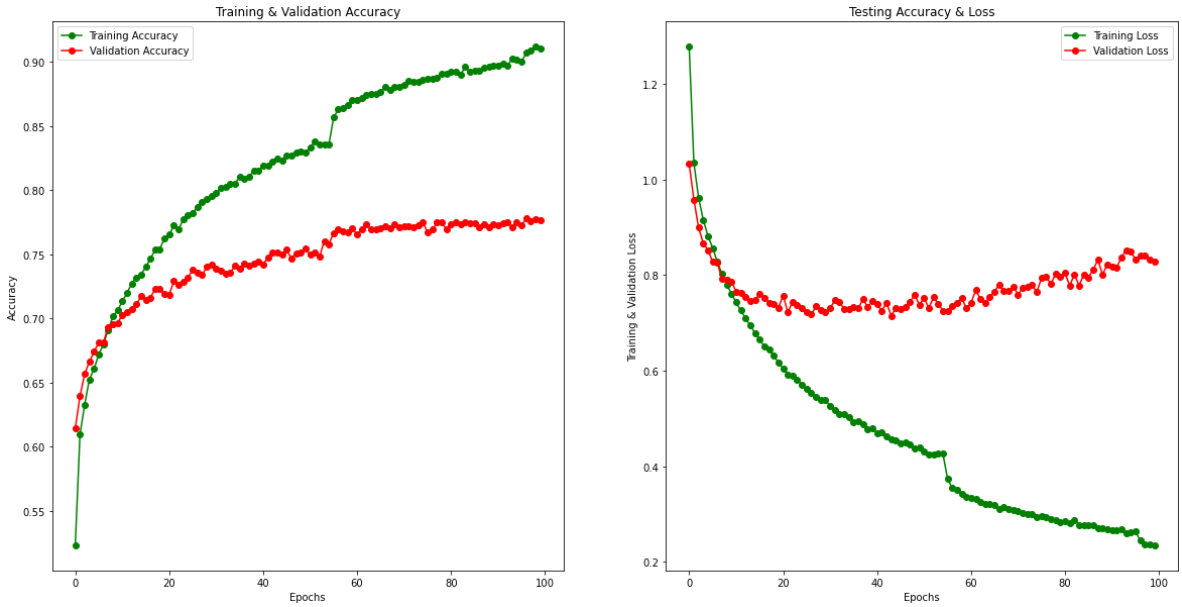


Figura 52 Gráficas de Accuracy y Loss durante el entrenamiento del cuarto modelo de la tabla 11.

El patrón mostrado en las gráficas es el que adoptan los primeros modelos, hasta que se decide probar un enfoque de extracción de características más simple. Lo que más preocupaba era el comportamiento del *Loss*, ya que sugería la posibilidad de que el modelo estaba *overfitting*. El *overfitting* es una de las mayores amenazas para estos algoritmos de clasificación, por lo tanto, se consideraba importante tenerlo en el punto de mira.

Para solucionar este problema se puso el foco en probar diferentes modificaciones en la estructura y en el ajuste de hiperparámetros.

Cambios en los métodos y en la estructura:

1. Cambiar la estrategia de Feature Extraction, considerando que la extracción de los MFCC va a ser suficiente para este trabajo. Esto reduce la carga computacional y preserva mejor la forma original de los datos, lo cual resulta ser la mejor opción que seguir.
2. Cambiar las Pooling Layers de Max. a Avg., la primera lleva a cabo agrupaciones más discretas, al estar realizando la operación máxima, extrae las características más destacadas, como los bordes, de cada audio. Average Pooling realiza cálculos más

generalizados. Las características MFCC aportan mucha información sobre las características más destacadas, y durante las primeras fases de entrenamiento se puede comprobar que una extracción de características excesiva puede dañar los datos y, por lo tanto, el modelo.

3. Aplicar una arquitectura hibrida CNN-LSTM, sacando partido de los beneficios de ambas modalidades.

Ajuste de hiperparámetros (Hyperparemeter Tuning)

Learning rate. La tasa de aprendizaje es un hiperparámetro muy importante y, a menudo, requiere algo de experimentación. Con una tasa de aprendizaje demasiado grande, puede rebotar alrededor de un óptimo, o puede comenzar disparando hacia una parte del espacio de parámetros donde los gradientes desaparecen. Con un valor demasiado pequeño, es posible que tarde demasiado en converger a un óptimo, o puede que encuentre un óptimo local deficiente. Estos efectos se ven atenuados por el efecto impulso del optimizador Adam.

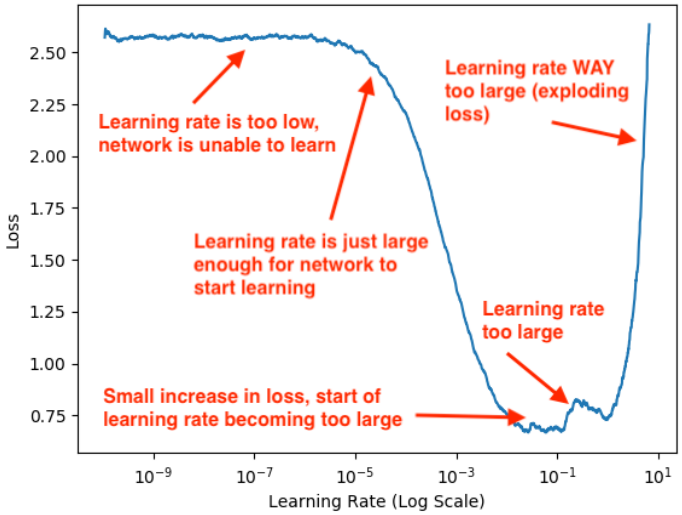


Figura 53 Comportamiento del Loss en función del Learning Rate (Rosebrock, 2021)

Es posible que la causa de que el *Loss* incremente al final durante el entrenamiento venga de que el *learning rate* sea demasiado alto (figura 55) [73]. Dicho esto, se decide trabajar con el *learning rate* por defecto que establece Adam, y luego implementar la retro llamada *ReduceLROnPlateau*, que soluciona este problema reduciendo el *learning rate* cuando la métrica objetivo deja de mejorar.

Aplicar estos cambios tiene un impacto notable en las curvas de resultados del entrenamiento. Los mejores resultados pertenecen el modelo femenino, seguido por el masculino y el mixto. Esto sugiere que una división por sexo es beneficioso para esta tarea de clasificación de emociones.

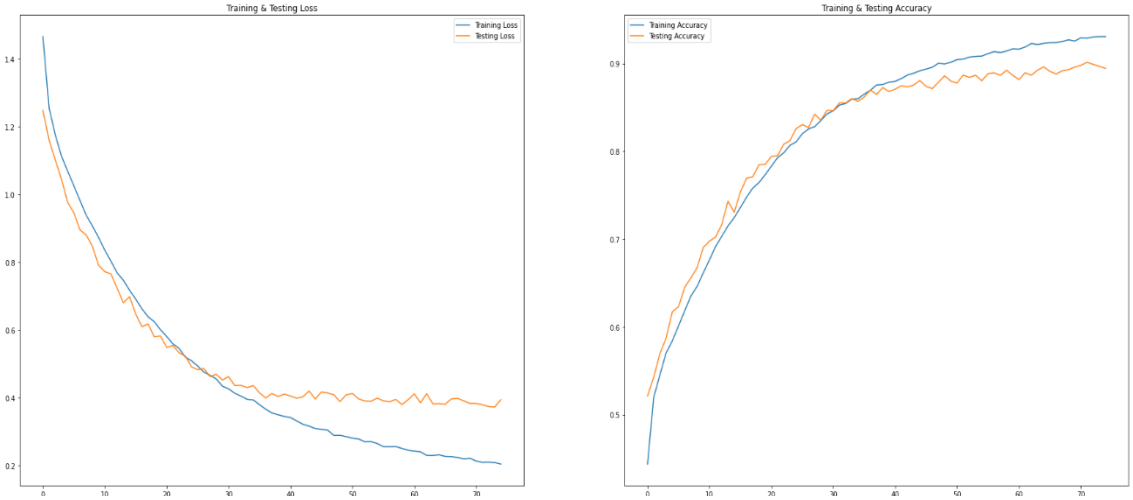


Figura 54 Curvas de Accuracy y Loss para el entrenamiento y validación del modelo final. Género mixto

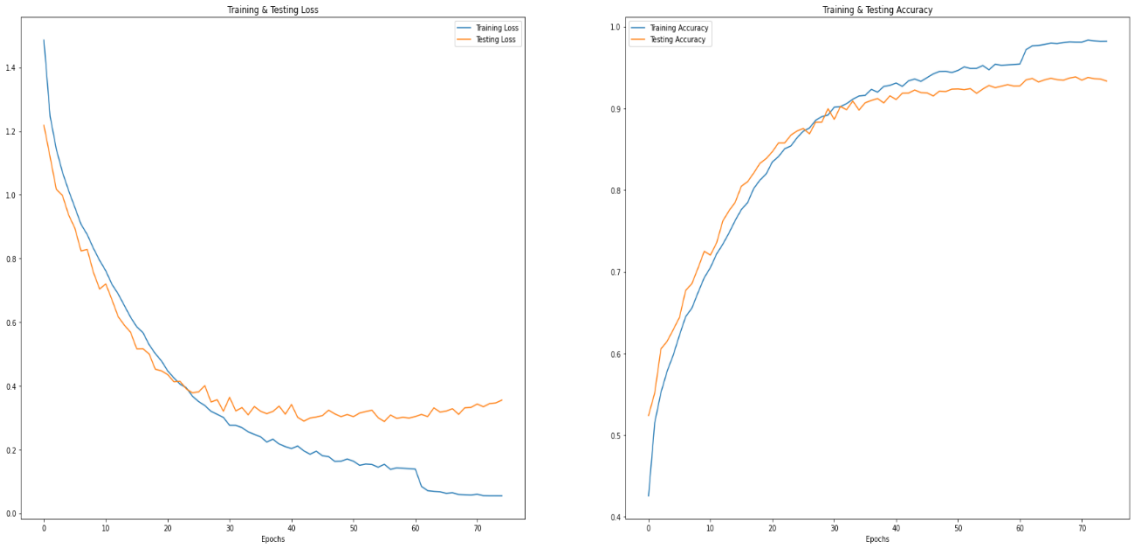


Figura 55 Curvas de Accuracy y Loss para el entrenamiento y validación del modelo final. Género femenino

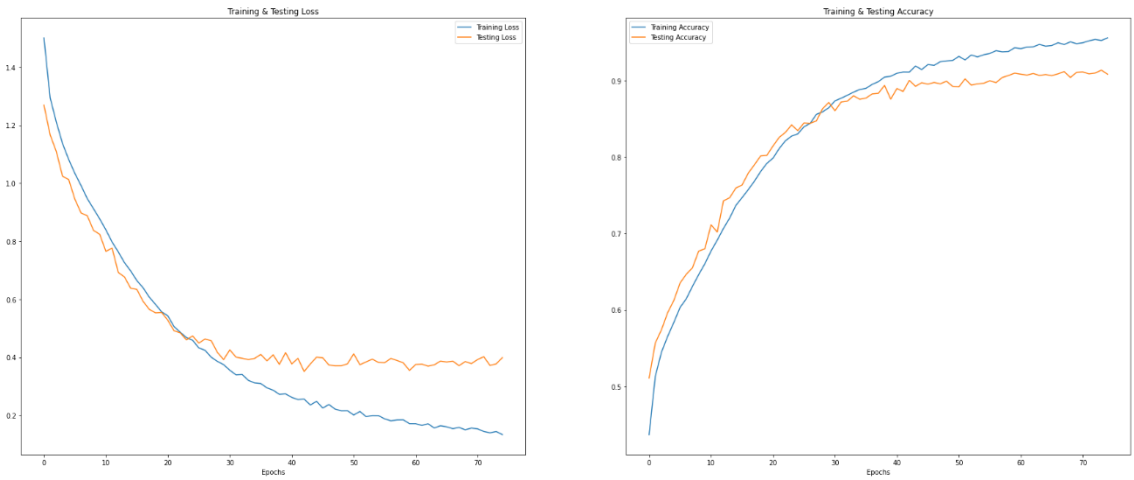


Figura 56 Curvas de Accuracy y Loss para el entrenamiento y validación del modelo final. Género masculino

Una vez se tiene la arquitectura que ofrece mayor eficacia, se pasa a la fase de testeo, durante la cual se analiza en detalle la capacidad de predicción del modelo seleccionado. Para ello se analizan las matrices de confusión, a través de las cuales se identifica si existen confusiones específicas en la detección de clases.

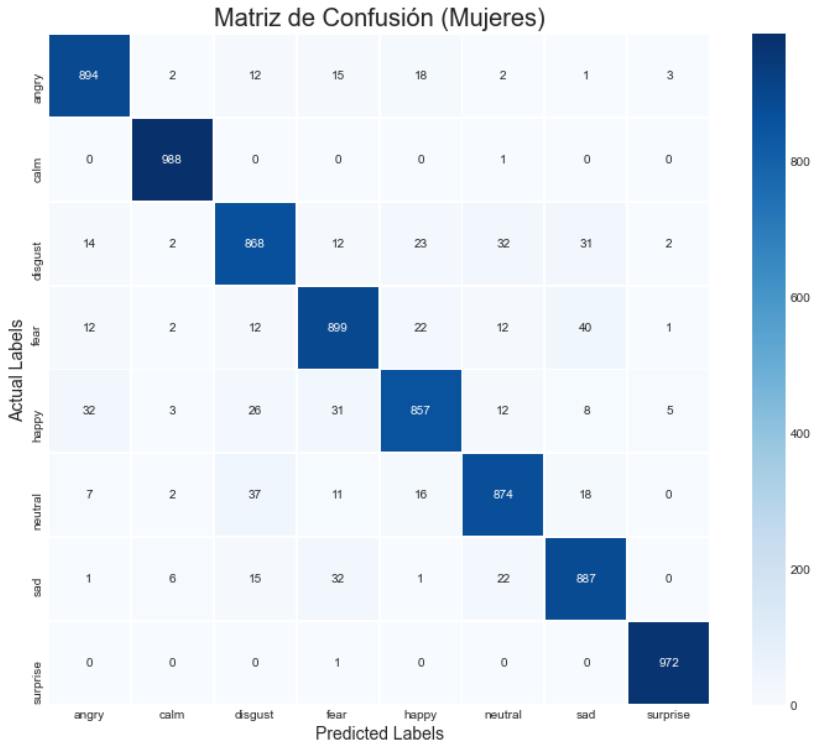


Figura 57 Matriz de confusión para las predicciones de emociones del dataset femenino

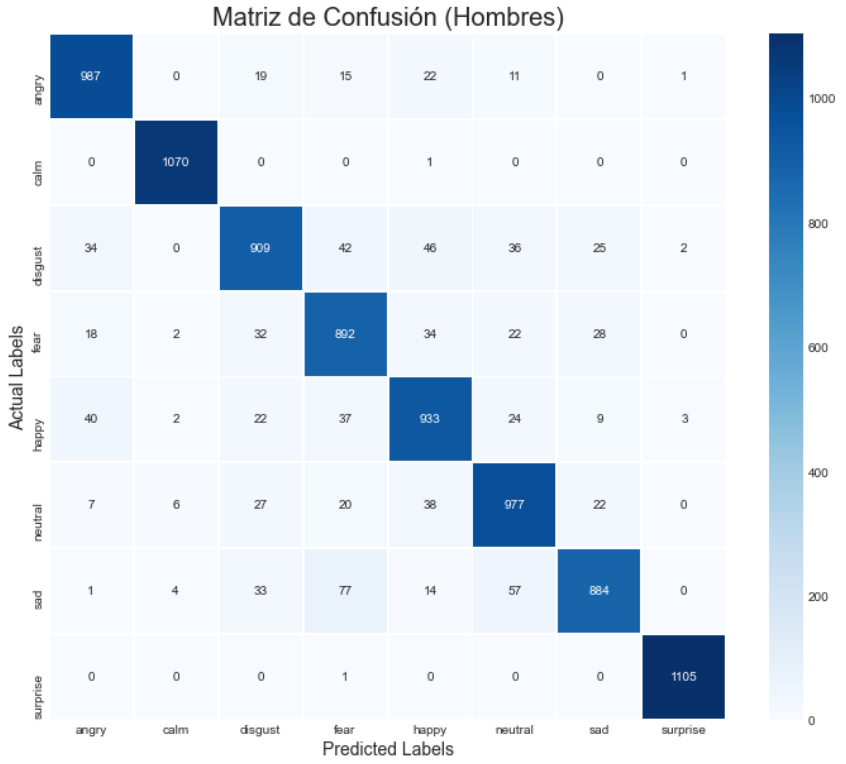


Figura 58 Matriz de confusión para las predicciones de emociones del dataset masculino

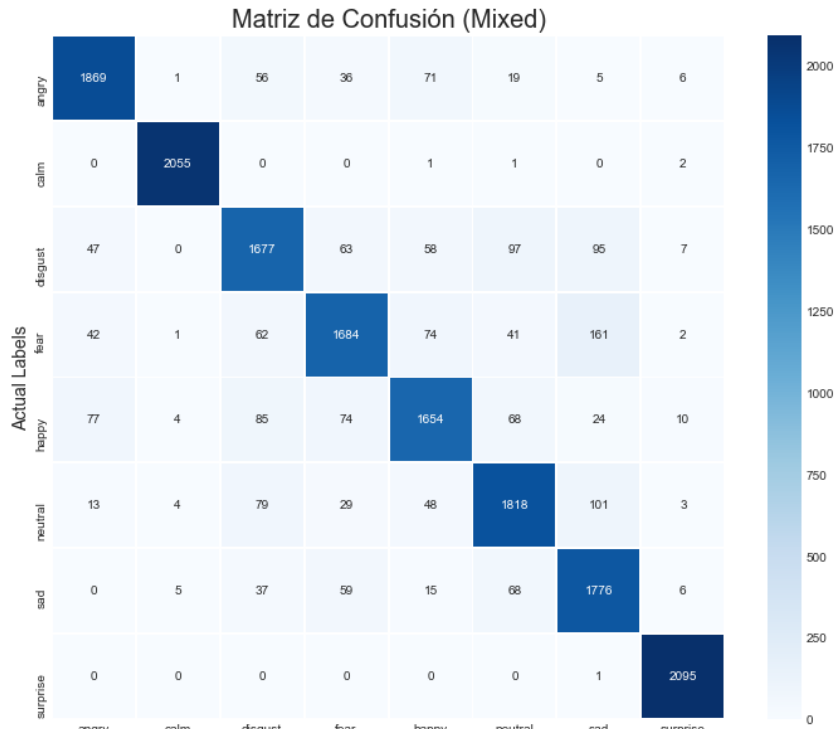


Figura 59 Matriz de confusión para las predicciones de emociones del dataset mixto

Por lo general, el clasificador realiza predicciones exitosas. Se observa que las emociones que más confunde son *fear* con *sad,* lo cual resulta una confusión razonable. El patrón es muy similar para las tres matrices de confusión, lo que indica que el desempeño del algoritmo no va a ser condicionado por el género del interlocutor.

Las emociones que mejor se clasifican son *surpise* y *calm,* pero no existe una diferencia muy notable entre estas y el resto. El ligero aumento en estas dos categorías puede ser debido a que son las dos emociones a las cuales se le aplica el *oversampling* al comienzo. Esta técnica produce muestras menos variadas de los datos, lo que hace que estas dos categorías tengan datos a partir de los cuales es más fácil generalizar.

La infraestructura final del sistema es capaz de recibir peticiones de múltiples usuarios. Se tiene que replicar el proceso realizado de extracción de característica y procesado de datos para poder ser servidos al modelo entrenado. La figura 62 plasma una imagen global de este sistema.

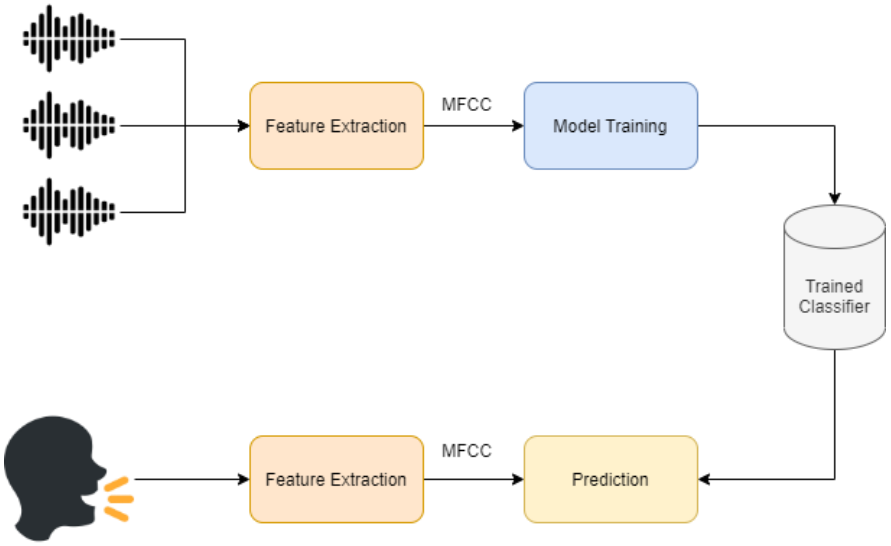


Figura 60 Sistema de reconocimiento de emociones

Aplicación web

La aplicación web final esta compuesta por los siguientes componentes:

Página principal:

- Para grabar la voz:
  - Botón Record (grabar más o menos 5 s de audio)
  - Botón Stop
- Visualización del audio generado: posibilidad de escuchar, guardar y subir al servidor
- Para realizar la predicción es necesario subir la grabación al servidor a través del enlace Upload

- Selección de género con un drop-down
- Botón Go para redirigir a la página de predicciones

Páginas de predicciones:

- Muestra la emoción detectada con el diseño adaptado a cada emoción

El logo aparece en la página principal y como icono favicon.

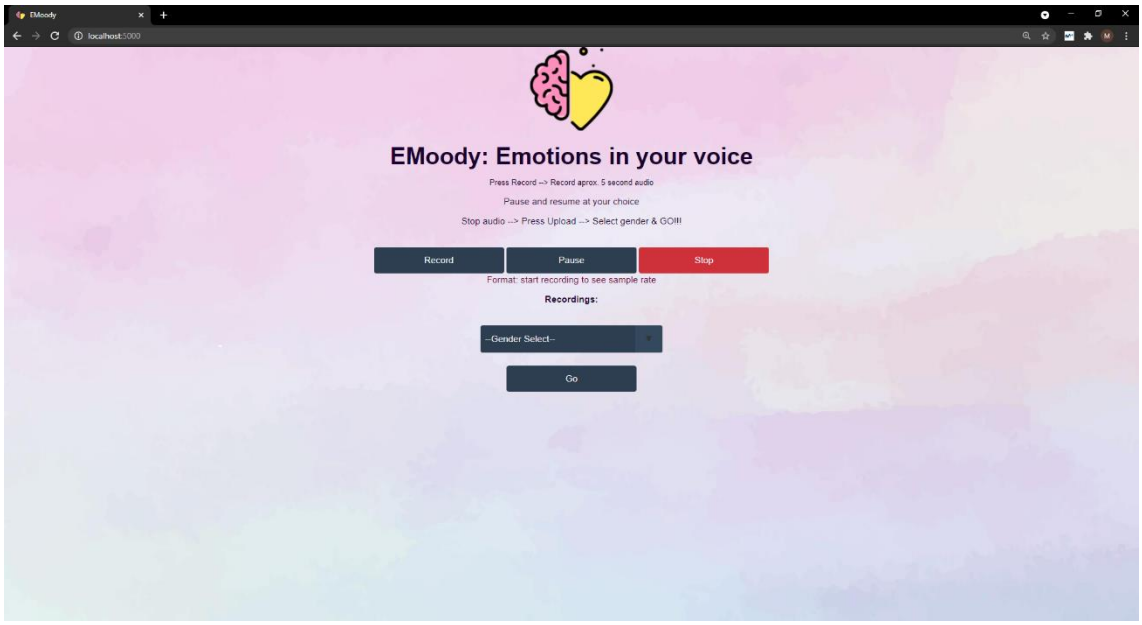


Figura 61 Pantalla principal de la aplicación web EMoody

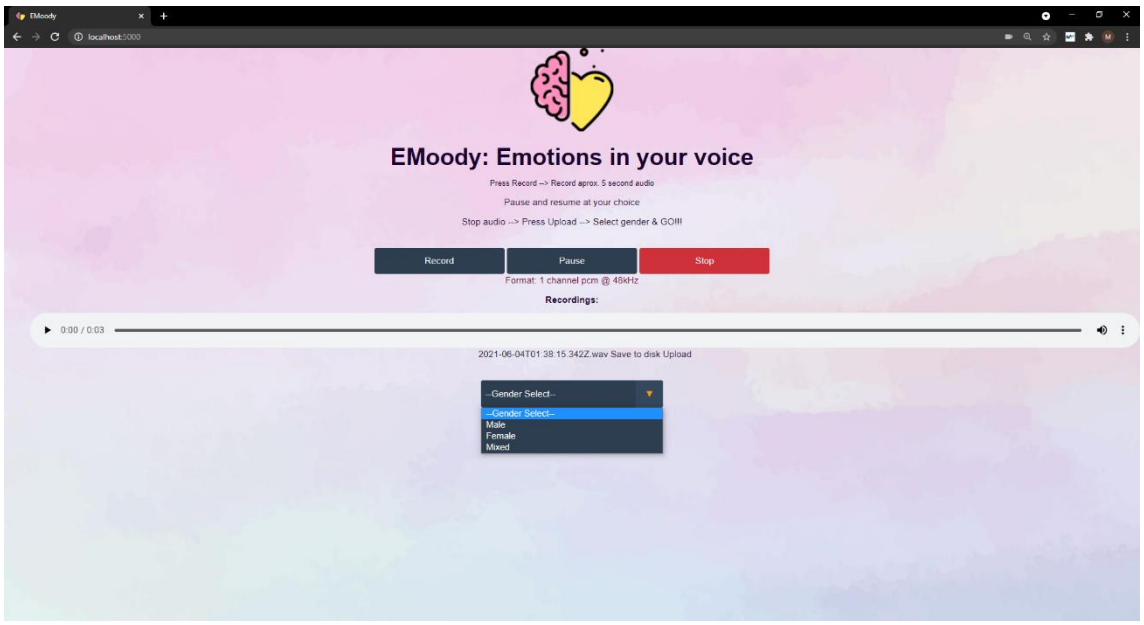


Figura 62 Pantalla principal después de haber grabado un audio y mostrando el dropdown para seleccionar género

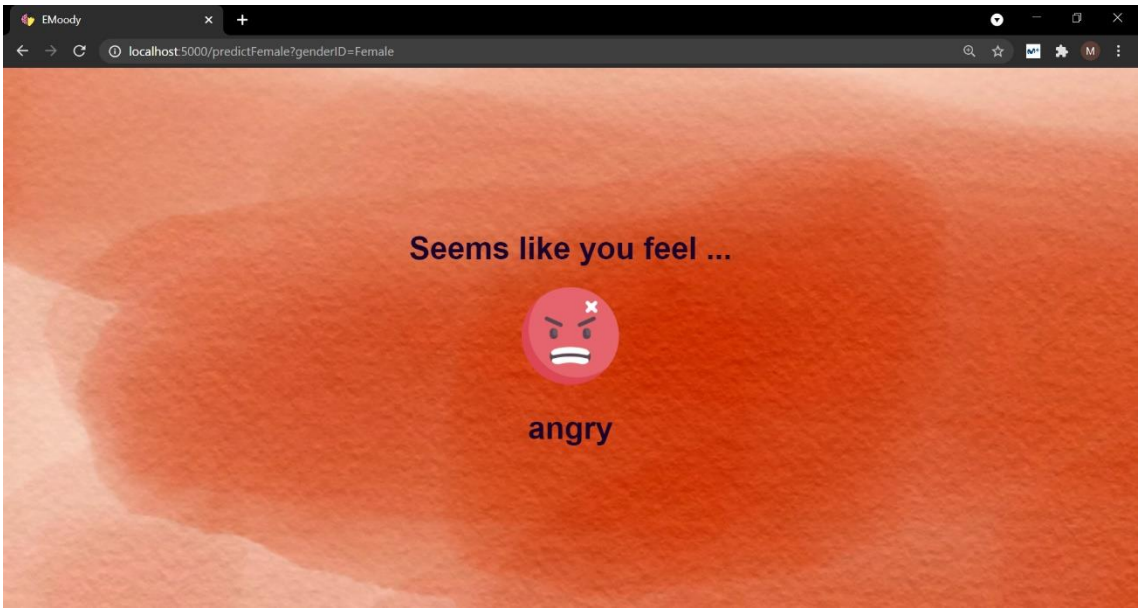


Figura 63 Pantalla de predicción- enfado

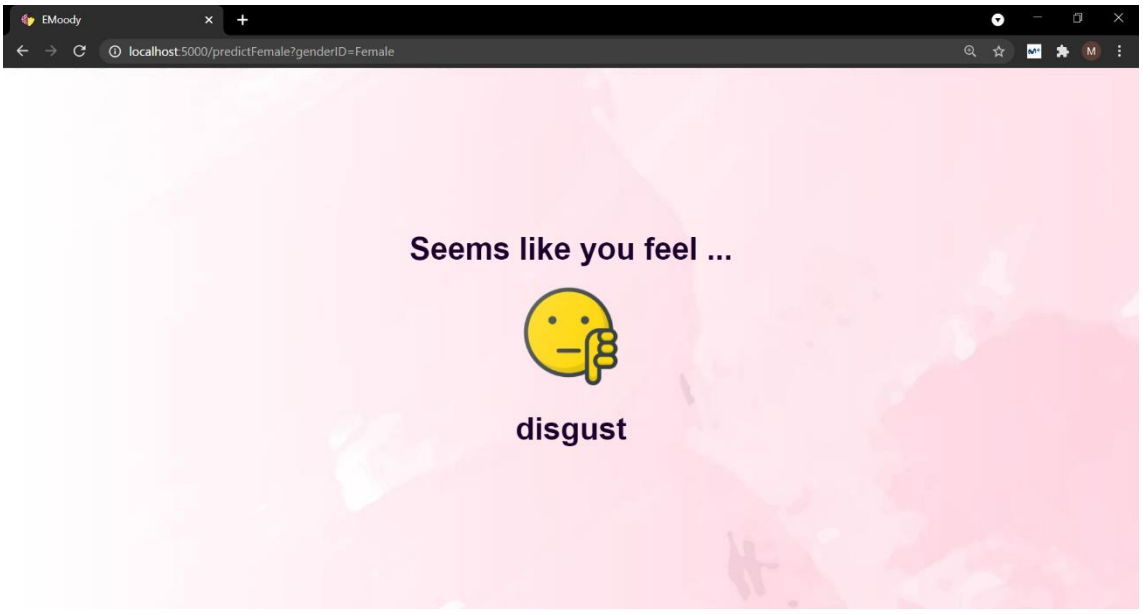


Figura 64 Pantalla de predicción- descontento/ repugnado

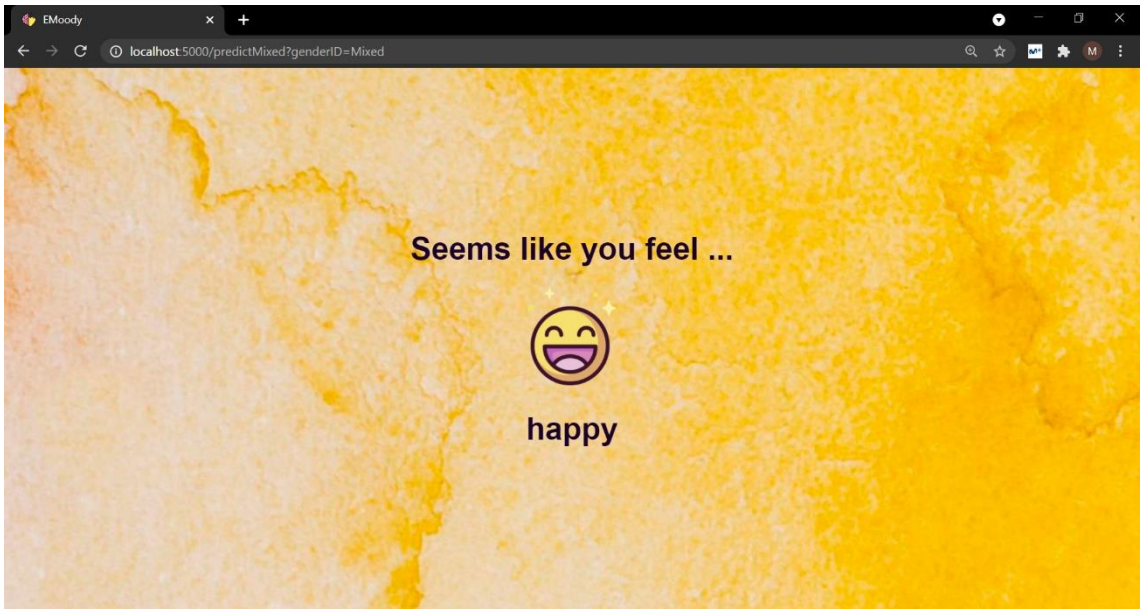


Figura 65 Pantalla de predicción- feliz

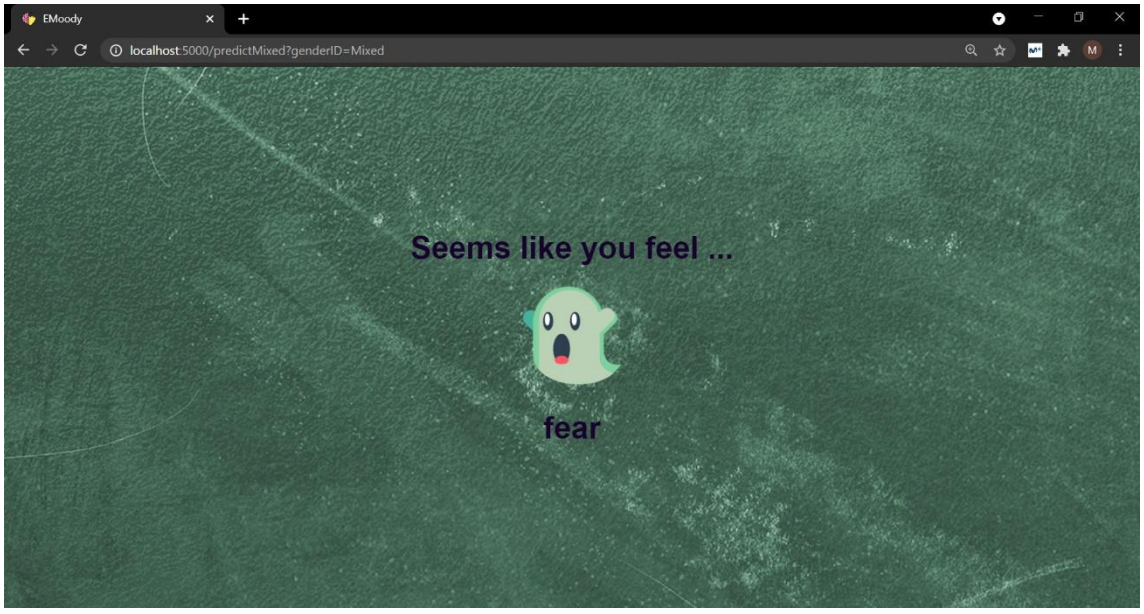


Figura 66 Pantalla de predicción- miedo

Existe la posibilidad de poder volver atrás para repetir el proceso, o refrescar la página en el caso de querer una segunda predicción.

Capítulo 4. DISCUSIÓN

Este proyecto comenzó con el objetivo de entender como implementar redes de aprendizaje profundo y como conseguir una arquitectura que realice las tareas de predicción de manera más eficaz. Sin embargo, a medida que transcurría el curso del proyecto, se volvió predominante la importancia de la ingeniería de características. Se observa que, en el reconocimiento de emociones a través de la voz, la elección de las características extraídas del audio tiene un mayor impacto que la complejidad del modelo. Incluso modelos más simples pueden devolver resultados más precisos.

Uno de los mayores obstáculos que se ha presentado es la limitación de datos de entrenamiento. Un dataset de entrenamiento mayor hubiese tenido un impacto muy positivo en los resultados, pero el poder de cómputo era limitado a la hora de realizar el entrenamiento. La falta de fuentes de datos multi lingues también es notable; hubiese sido especialmente interesante poder trabajar con un dataset grande de audios en castellano. Sin embargo, los datos empleados son muy completos y variados para realizar una prueba de concepto. En un futuro se pueden emplear tecnologías más avanzadas que permitan entrenamientos a mayor escala.

La obtención de algoritmos ha requerido de un sistema de acciones previo que incluye un pre procesado de datos que extrae las características que se han considerado importantes para caracterizar cada emoción. El proceso de extracción de características añade robustez al modelo, que a su vez hace su propia labor de extracción de características dentro de la red neuronal.

Mediante el entrenamiento de un modelo de aprendizaje profundo, se alcanza un 93% de eficacia en datos de testeo para uno de los modelos, y un 90% y 89% respectivamente para los otros. Estos resultados son muy positivos, pero, sin embargo, se ha comprobado que uno de los obstáculos a los que se enfrenta este sistema a la hora de la puesta en marcha en producción, es que su capacidad de respuesta ante nuevas fuentes de datos es más pobre. Esto se debe a la alta variedad de sistemas de recogida de voz, lo que saca a los modelos de su zona de confort. Por ello, una de las incorporaciones a futuro es el reentreno del modelo cada vez que reciba datos nuevos.

El *framework* Flask ha ofrecido todas las funcionalidades que se buscaban para poder darle una aplicabilidad a los modelos desde el punto de vista de interacción con usuario. Es una herramienta sencilla que ha permitido hacer de un código complejo de aprendizaje automático una proyección visual en el navegador. Sin embargo, por si solo puede resultar ser una tarea tediosa configurar el entorno de trabajo para su despliegue, por lo que en el caso de querer poner en producción este proyecto sería fundamental un hospedaje en la nube.

Capítulo 5. CONCLUSIONES

5.1 Conclusiones del trabajo

Cada vez se tienen más en cuenta las emociones para explicar el comportamiento y las enfermedades neurológicas en el ser humano. El resultado de este proyecto, una herramienta basada en inteligencia artificial capaz de detectar el estado emocional de una persona, se ve como un complemento de gran utilidad en muchas áreas. De especial importancia son las relacionadas con la salud mental, ya que este desarrollo puede complementar y ayudar en el diagnóstico de enfermedades.

Cuando se comenzó a trabajar en este proyecto, se tenía una idea de la potencia detrás de los algoritmos de Deep Learning, pero durante la elaboración se ha podido confirmar su fortaleza resolviendo problemas complejos. Se ha comprobado que no es necesario construir una estructura compleja, sino que lo importante es cuidarlo dándole la información limpia y adecuada y ajustando los diferentes parámetros que necesita para que se sienta cómodo y pueda optimizar sus tareas. Se considera que emplear redes neuronales artificiales para tratar de descifrar el funcionamiento de las naturales es el método más adecuado.

El añadido final es ofrecer una aplicación con una interfaz de usuario adecuada para que este proceso sea automatizado y pueda devolver resultados de manera interactiva y visual. El proyecto ha conseguido de manera exitosa desarrollar el motor de reconocimiento y el diseño de una interfaz atractiva para el usuario.

5.2 Conclusiones personales

Realizar este proyecto ha supuesto un reto tanto académico como personal. El área del Deep Learning lleva atrayendo mi curiosidad desde que empecé la carrera en este grado. Es impresionante como esta área de estudio ha avanzado tan rápido en los últimos años y poder realizar un trabajo que ha requerido estudiar profundamente este tema mientras está en auge ha sido muy enriquecedor.

A su vez, he tenido la oportunidad de adentrarme en el mundo del desarrollo de aplicaciones. Mis especialidades siempre han estado orientadas al desarrollo de los componentes back-end, por lo que poder nutrir habilidades en el campo del desarrollo de sistemas front-end ha sido un complemento muy grato para mis estudios. A su vez, ha resultado ser divertido y agradable trabajar en un área más orientado hacia técnicas de diseño, ¡y dejar el picar código de lado un rato!

Trabajar en la fusión de las dos dinámicas que personifican este proyecto me ha dado la oportunidad de aprender y profundizar sobre las dos ramas de estudio que más disfruto. Sin duda, este proyecto es la semilla hacia el futuro profesional al que me inclino.

Capítulo 6. FUTURAS LÍNEAS DE TRABAJO

Esencialmente lo más interesante sería hacer accesible esta aplicación al público. Por el momento solo es posible hacer un despliegue en local y accesible desde localhost del propio dispositivo. Por lo tanto, el próximo paso sería desplegar la aplicación web de Flask en un servidor con disponibilidad al público. La opción más llamativa es una implementación utilizando servicios de la nube. Las opciones más destacadas de desarrollo web con Flask es integración con Heroku, Google Cloud Platform (GCP) o Amazon Web Services (AWS). El Cloud Computing tiene numerosas ventajas, la más beneficiosa para este proyecto es la posibilidad de lanzar la aplicación y que se hospede en remoto, ahorrando la necesidad de un dispositivo hardware pendiente de dar respuesta a las peticiones. La gestión de peticiones es otra de las grandes ventajas; estas pueden llegar en grandes cantidades, sin perder la rapidez y eficacia del sistema de respuesta. A su vez supondría un ahorra en costes hardware, ya que los servicios en la nube funcionan bajo un modelo de pago por uso, lo que optimizaría los recursos tanto computacionales como económicos.

Una vez la aplicación estuviese alojada en la nube, se consideraría ampliar el abanico de plataformas a las que apuntar. Una de las opciones desde el comienzo ha sido desarrollar una aplicación móvil multi-plataforma hospedada en la nube y disponible desde el Play Store y Apple Store desde cualquier smartphone, cuyo uso es aun más generalizado que el de un ordenador.

Para potenciar la eficacia del algoritmo en sí, un hosting en la nube permitiría una rutina de entrenamiento constante. Es decir, cada dato entrante puede entrar en un bucle de retroalimentación y además de devolver una predicción en base a él, el algoritmo podría reentrenarse continuamente. Así, podrá aprender en base a fuentes de datos variados, siendo la carencia de esto una de las grandes debilidades del sistema actual. A su vez, sería de interés realizar entrenamientos con datos de múltiples idiomas para hacer modelos adecuados a los diferentes mercados, no solo el de habla inglesa.

En cuanto a experiencia de usuario, el foco está en desarrollar una UX más personalizada. A través de un log in el usuario podría acceder a su propia área de usuario con información sobre sus peticiones y su historial. Esto conllevaría incluir un componente primordial de seguridad y privacidad, incluyendo la administración de cookies. La aplicación desarrollada en el presente trabajo no requiere aplicar estas políticas ya que no es accesible al público y su presencia solo esta en local por el momento.

La visión de hacer de este trabajo un proyecto a futuro se ha desarrollado a medida avanzaba este trabajo. Sin duda, poder hacer de este trabajo de fin de grado un producto real con presencia en el mercado, bajo la marca diseñada, es una idea muy cautivadora.

Capítulo 7. REFERENCIAS

[1] Affective Computing Market Outlook with COVID-19 Impact Analysis & Opportunities, Future Challenges, Growth Statistics and Forecast to 2023 – Factory Gate. (2021). Disponible en: https://factorygate.co.uk/uncategorized/997096/affective-computing-market-outlook-withcovid-19-impact-analysis-opportunities-future-challenges-growth-statistics-and-forecast-to2023/ (Consultado el 14 de enero, 2021).

[2] Brownlee, J. (2021). Your First Deep Learning Project in Python with Keras Step-By-Step. Disponible en: https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/ (Consultado el 21 de mayo, 2021)

[3] Brownlee, J. (2021). Why Is Imbalanced Classification Difficult?. Disponible en: https://machinelearningmastery.com/imbalanced-classification-is-hard/ (Consultado el 11 de marzo, 2021)

[4] Jung, H. W., Seo, Y. H., Ryoo, M. S., & Yang, H. S. (2004, November). Affective communication system with multimodality for a humanoid robot, AMI. In *4th IEEE/RAS International Conference on Humanoid Robots, 2004.* (Vol. 2, pp. 690-706). IEEE.

[5] Luneski, A., Konstantinidis, E., & Bamidis, P. (2010). Affective medicine: a review of affective computing efforts in medical informatics. *Methods of information in medicine*, *49*(3), 207-218.

[6] Mesko, B. Artificial Intelligence Will Redesign Healthcare. 2016. Disponible en: https://www.linkedin.com/pulse/artificial-intelligence-redesign-healthcare-bertalanmesk%C3%B3-md-phd (Consultado el 8 de enero, 2021)

[7] Google Health. (2021). Disponible en: https://health.google/health-research/ (Consultado el 8 de enero, 2021).

[8] Market, A. (2021). Affective Computing Market Size, Share and Global Forecast to 2024 | COVID-19 Impact Analysis | MarketsandMarkets. Disponible en: https://www.marketsandmarkets.com/Market-Reports/affective-computing-market-130730395.html (Consultado el 20 de marzo, 2021)

[9] Schuller, B. W. (2018). Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends. *Communications of the ACM*, *61*(5), 90-99.

[10] France, D. J., Shiavi, R. G., Silverman, S., Silverman, M., & Wilkes, M. (2000). Acoustical properties of speech as indicators of depression and suicidal risk. *IEEE transactions on Biomedical Engineering*, *47*(7), 829-837.

[11] Ingale, A. B., & Chaudhari, D. S. (2012). Speech emotion recognition. *International Journal of Soft Computing and Engineering (IJSCE)*, *2*(1), 235-238.

[12] Picard, R. W. (2000). *Affective computing*. MIT press.

[13] Piórkowska, Magda & Wrobel, Monika. (2017). Basic Emotions. 10.1007/978-3-319-28099-8_495-1.

[14] Clynes, M. (1977). *Sentics: The touch of emotions*. Anchor Press.

[15] Donaldson, M. (2017). Plutchik's wheel of emotions—2017. Update.

[16] Murugan, Harini. (2020). Speech Emotion Recognition Using CNN. International Journal of Psychosocial Rehabilitation. 24. 10.37200/IJPR/V24I8/PR280260.

[17] Ruiz, P., Apud, I., Maiche, A., González, H., Pires, A. C., Carboni, A., ... & González Perilli, F. (2016). Manual de introducción a la psicología cognitiva.

[18] Koolagudi, S. G., & Rao, K. S. (2012). Emotion recognition from speech: a review. *International journal of speech technology*, *15*(2), 99-117.

[19] Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H., & Alhussain, T. (2019). Speech emotion recognition using deep learning techniques: A review. *IEEE Access*, *7*, 117327-117345.

[20] Sithara, A., Thomas, A., & Mathew, D. (2018). Study of MFCC and IHC feature extraction methods with probabilistic acoustic models for speaker biometric applications. *Procedia computer science*, *143*, 267-276.

[21] Venkataramanan, K., & Rajamohan, H. R. (2019). Emotion recognition from speech. *arXiv preprint arXiv:1912.10458*.

[22] Akçay, M. B., & Oğuz, K. (2020). Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Communication*, *116*, 56-76.

[23] Lee, C. M., & Narayanan, S. S. (2005). Toward detecting emotions in spoken dialogs. *IEEE transactions on speech and audio processing*, *13*(2), 293-303.

[24] Nwe, T. L., Foo, S. W., & De Silva, L. C. (2003). Speech emotion recognition using hidden Markov models. *Speech communication*, *41*(4), 603-623.

[25] Anagnostopoulos, C. N., Iliou, T., & Giannoukos, I. (2015). Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011. *Artificial Intelligence Review*, *43*(2), 155-177.

[26] Logan, B. (2000, October). Mel frequency cepstral coefficients for music modeling. In *Ismir* (Vol. 270, pp. 1-11).

[27] Menos es más - Wikipedia, la enciclopedia libre. (2021). Disponible en: https://es.wikipedia.org/wiki/Menos_es_m%C3%A1s#:~:text=%C2%ABMenos%20es%20m%C3%A1s%C2%BB%20(less,movimiento%20art%C3%ADstico%20conocido%20como%20minimalismo. (Consultado el 10 de mayo, 2021)

[28] Briega, R. (2021). Ejemplo de Machine Learning con Python - Selección de atributos. Disponible en: https://relopezbriega.github.io/blog/2016/04/15/ejemplo-de-machine-learning-con-python-seleccion-de-atributos/ (Consultado el 10 de mayo, 2021)

[29] Omuya, E. O., Okeyo, G. O., & Kimwele, M. W. (2021). Feature Selection for Classification using Principal Component Analysis and Information Gain. *Expert Systems with Applications*, *174*, 114765.

[30] F. Song, Z. Guo and D. Mei, "Feature Selection Using Principal Component Analysis," *2010 International Conference on System Science, Engineering Design and Manufacturing Informatization*, 2010, pp. 27-30, doi: 10.1109/ICSEM.2010.14.

[31] J. Nobre, F. Neves. Combining Principal Component Analysis, Discrete Wavelet Transform and XGBoost to trade in the financial markets. Expert Systems with Applications., 125 (1) (2019), pp. 181-194

[32] Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H., & Alhussain, T. (2019). Speech emotion recognition using deep learning techniques: A review. *IEEE Access*, *7*, 117327-117345.

[33] Yu, D., & Deng, L. (2016). *AUTOMATIC SPEECH RECOGNITION*. Springer london limited.

[34] Huang, Z., Dong, M., Mao, Q., & Zhan, Y. (2014, November). Speech emotion recognition using CNN. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 801-804).

[35] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117.

[36] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117.

[37] Different between CNN, RNN（Quote）. (2021). Disponible en: https://medium.com/@Aj.Cheng/different-between-cnn-rnn-quote-7c224795db58 (Consultado el 21 de enero, 2021)

[38] Memory, E. (2021). Long Short Term Memory | Architecture Of LSTM. Disponible en: https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/ (Consultado el 20 de febrero, 2021)

[39] Islam, M. Z., Islam, M. M., & Asraf, A. (2020). A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Informatics in medicine unlocked*, *20*, 100412.

[40] How to Create an API Using The Flask Framework | Nordic APIs |. (2021). Disponible en: https://nordicapis.com/how-to-create-an-api-using-the-flask-framework/ (Consultado el 1 de mayo, 2021)

[41] ¿Qué es un WSGI?. (2021). Disponible en: https://medium.com/@nachoad/que-es-wsgi-be7359c6e001 (Consultado el 11 de mayo, 2021)

[42] Cómo crear una aplicación Web usando Flask en Python 3 | DigitalOcean. (2021). Disponible en: https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3-es (Consultado el 11 de mayo, 2021)

[43] Vocal Emotion Recognition Test by Empath. (2021). Disponible en: https://webempath.net/lp-eng/ (Consultado el 21 de mayo, 2021)

[44] Fabien, M. (2021). Emotion Recognition WebApp. Disponible en: https://maelfabien.github.io/project/poleemploi/#real-time-multimodal-emotion-recognition (Consultado el 21 de mayo, 2021)

[45] Vmote - Voice Messaging. (2021). Disponible en: https://apps.apple.com/gt/app/vmote-voice-messaging/id1273369846 (Consultado el 21 de mayo, 2021)

[46] Adam, F: The Genuine Works of Hippocrates, Translated from the Greek with a Preliminary Discourse and Annotations. New York, W Wood and Co (1886)

[47] The World Health Organisation (WHO) (1948). Preamble to the constitution of the World Health Organisation as adopted by the International Health Conference. Disponible en: https://apps.who.int/gb/bd/PDF/bd47/SP/constitucion-sp.pdf?ua=1 (Consultado el 5 de enero, 2021).

[48] Gordon, J. S. (1981). Holistic medicine: toward a new medical model. *The Journal of clinical psychiatry*, *42*(3), 114-119.

[49] de Silva, P. (2020). Health and Emotional Experience. In *Mindfulness-based Emotion Focused Counselling* (pp. 171-176). Palgrave Macmillan, Cham.

[50 lama] Goleman, D. (Ed.). (2003). *Healing emotions: Conversations with the Dalai Lama on mindfulness, emotions, and health*. Shambhala publications.

[51] Cohen, S., Tyrrell, D. A., & Smith, A. P. (1991). Psychological stress and susceptibility to the common cold. *New England journal of medicine*, *325*(9), 606-612.

[52] Stone, A. A., Cox, D. S., Valdimarsdottir, H., Jandorf, L., & Neale, J. M. (1987). Evidence that secretory IgA antibody is associated with daily mood. *Journal of personality and social psychology*, *52*(5), 988.

[53] Mesko, B. Artificial Intelligence Will Redesign Healthcare. 2016. Disponible en: https://www.linkedin.com/pulse/artificial-intelligence-redesign-healthcare-bertalan-mesk%C3%B3-md-phd (Consultado el 8 de enero, 2021).

[54] Google Health. (2021). Disponible en: https://health.google/health-research/ (Consultado el 8 de enero, 2021).

[55] Salud Mental y COVID-19 - OPS/OMS | Organización Panamericana de la Salud. (2021). Disponible en: https://www.paho.org/es/salud-mental-covid-19 (Consultado el 19 de mayo, 2021)

[56] de Las Heras-Pedrosa C, Sánchez-Núñez P, Peláez JI. Sentiment Analysis and Emotion Understanding during the COVID-19 Pandemic in Spain and Its Impact on Digital Ecosystems. Int J Environ Res Public Health. 2020 Jul 31;17(15):5542

[57] Statement on the fifth meeting of the International Health Regulations (2005) Emergency Committee regarding the coronavirus disease (COVID-19) pandemic. (2021). Disponible en: https://www.who.int/news/item/30-10-2020-statement-on-the-fifth-meeting-of-the-international-health-regulations-(2005)-emergency-committee-regarding-the-coronavirus-disease-(covid-19)-pandemic (Consultado el 19 de mayo, 2021)

[58] Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391.

[59] Cao, H., Cooper, D. G., Keutmann, M. K., Gur, R. C., Nenkova, A., & Verma, R. (2014). CREMA-D: Crowd-sourced Emotional Multimodal Actors Dataset. *IEEE transactions on affective computing*, *5*(4), 377–390. https://doi.org/10.1109/TAFFC.2014.2336244

[60] Machine learning demystified: the importance of data. (2021). Disponible en: https://www.information-age.com/machine-learning-demystified-importance-data-123466738/ (Consultado el 21 de febrero, 2021)

[61] Jesmeen, M. Z. H., Hossen, J., Sayeed, S., Ho, C. K., Tawsif, K., Rahman, A., & Arif, E. M. H. (2018). A survey on cleaning dirty data using machine learning paradigm for big data analytics. *Indonesian Journal of Electrical Engineering and Computer Science*, *10*(3), 1234-1243.

[62] Brownlee, J. (2021). How to use Data Scaling Improve Deep Learning Model Stability and Performance. Retrieved 21 May 2021, from https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/

[63] Oversampling and Undersampling. (2021). Disponible en: https://towardsdatascience.com/oversampling-and-undersampling-5e2bbaf56dcf (Consultado el 10 de marzo, 2021)

[64] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

[65] How to Effortlessly Handle Class Imbalance with Python and SMOTE. (2021). Disponible en: https://towardsdatascience.com/how-to-effortlessly-handle-class-imbalance-with-python-and-smote-9b715ca8e5a7 (Consultado el 8 de marzo, 2021)

[66] franspg, V. (2021). Generación de datos artificiales (Data Augmentation). Disponible en: https://franspg.wordpress.com/2020/01/27/generacion-de-datos-artificiales-data-augmentation/ (Consultado el 1 de marzo, 2021)

[67] Data Augmentation for Audio. (2021). Disponible en: https://medium.com/@makcedward/data-augmentation-for-audio-76912b01fdf6 (Consultado el 1 de marzo, 2021)

[68] Yusnita, M. A., Paulraj, M. P., Yaacob, S., Yusuf, R., & Shahriman, A. B. (2013). Analysis of accent-sensitive words in multi-resolution mel-frequency cepstral coefficients for classification of accents in Malaysian English. *International Journal of Automotive and Mechanical Engineering*, *7*, 1053.

[69] Brownlee, J. (2021). Why One-Hot Encode Data in Machine Learning?. Disponible en: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/ (Consultado el 21 de marzo, 2021)

[70] Hamad, R. A., Yang, L., Woo, W. L., & Wei, B. (2020). Joint Learning of Temporal Models to Handle Imbalanced Data for Human Activity Recognition. *Applied Sciences*, *10*(15), 5293.

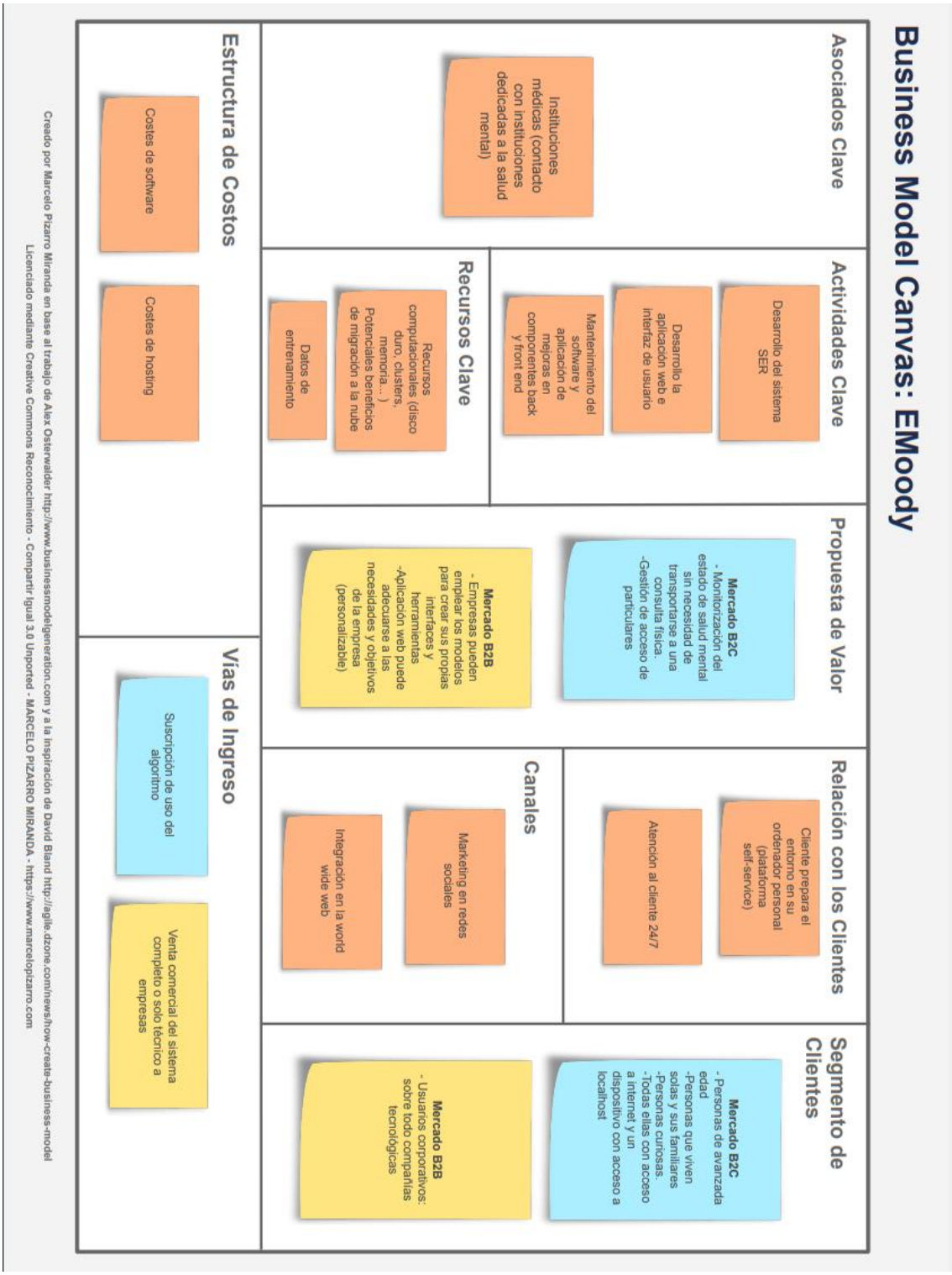
[71] Brownlee, J. (2021). Your First Deep Learning Project in Python with Keras Step-By-Step. Disponible https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/ (Consultado el 6 de abril, 2021)

[72] Brownlee, J. (2021). Why Is Imbalanced Classification Difficult?. Disponible en: https://machinelearningmastery.com/imbalanced-classification-is-hard/ (Consultado el 10 de marzo, 2021)

[73] Rosebrock, A. (2021). Keras Learning Rate Finder - PyImageSearch. Disponible en: https://www.pyimagesearch.com/2019/08/05/keras-learning-rate-finder/ (Consultado el 11 de febrero, 2021)

Capítulo 8. ANEXOS

Business Model Canvas: propuesta de modelo de negocio



[PÁGINA INTENCIONADAMENTE EN BLANCO]