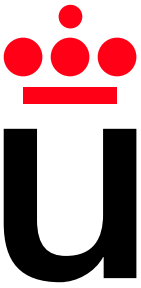


Universidad
Rey Juan Carlos

Escuela Técnica Superior
de Ingeniería Informática

Grado en Matemáticas

Curso 2023-2024

Trabajo Fin de Grado

DETECCIÓN DE FRAUDE DE VOZ A TRAVÉS DE MODELOS DE LA PARTE COMPLEJA DE SERIES DE FOURIER

Autor: Sara González Terroba
Tutores: Clara Simón De Blas, Jordi Luque Serrano

© 2023 Sara González Terroba
Algunos derechos reservados

Este documento se distribuye bajo la licencia "Atribución- CompartirIgual 4.0 Internacional" de Creative Commons, disponible en:
https://creativecommons.org/licenses/by-sa/4.0/deed.es

Agradecimientos

Siempre me ha fascinado la capacidad que tienen los autores para condensar y transmitir el agradecimiento y el cariño enorme hacia sus seres queridos en unas simples palabras, consiguiendo inundar al lector, completamente desconocido y ajeno, de algo curiosamente inefable.

Hoy ha llegado mi turno. Tanto que agradecer y tan pocas palabras.

Quiero comenzar agradeciendo a mis dos tutores, Clara y Jordi. La realización de este Trabajo Final de Grado no ha sido sencilla y comenzó en caminos que parecían llevar a destinos alejados de donde ha terminado desembocando.
Le agradezco a Clara cómo me tendió la mano cuando siquiera sabía cómo salir de la balsa en la que estaba hace unos meses, por motivarme y confiar en mí en todo momento. Tuve la suerte de que unos años atrás también fue mi profesora y, a día de hoy más que nunca, tengo claro que si tengo que pensar en las cualidades que hacen a un buen profesor, no hay forma de que no piense en las suyas.
Le agradezco a Jordi por aceptarme para trabajar en un reto real e interesante como es la detección y generación de voces falsas y por esperar de mí grandes cosas. Es un gran investigador y he de agradecerle a Telefónica por darme la oportunidad, mediante el programa Tutoría, de trabajar con él.
Gracias a ambos por cómo me han acompañado en mi aprendizaje, por el tiempo que me han dedicado, las innumerables reuniones y por todo lo que me han enseñado en este proceso. Tengo la certeza de que este trabajo es tan suyo como mío, ya que sin ellos no lo hubiera podido realizar.

Quisiera agradecer a Luis y a Pedro, ejemplos de lo que es ser un buen profesional y mejor persona. Trabajar junto a ellos ha sido una de las mejores cosas que he tenido el placer de vivir este año. Me han demostrado que el sueño de pertenecer a un equipo de personas apasionadas por aquello a lo que se dedican es una realidad muy posible y no lo dejaré de buscar durante todo mi futuro laboral.
El apoyo que me han proporcionado cada día a lo largo de estos doce meses que hemos pasado juntos ha sido fuente de motivación y esfuerzo en distintos ámbitos de mi vida, incluido este trabajo. Ha resultado ser de esas cosas que te marcan de por vida.

La siguiente mención que voy a hacer esperaba escribirle de otra forma, pero no consigo concretarla satisfactoriamente; así que aunque para el resto no tendrá tanto sentido, espero que, como siempre, ella me lea entre letras.
A Nazaret.

Finalmente, les dedico este trabajo a mi padre César, a mi madre Pilar y a mi hermana Alejandra. No puedo expresar realmente la gratitud que les tengo por estar los tres siempre a mi lado, día sí y día también. Por quererme incondicionalmente.
Les debo todo, pero en especial el grado en Matemáticas que con este Trabajo Fin de Grado cierro. No lo hubiera podido completar sin la seguridad que siempre me han dado. Por tenerles siempre respaldándome, para lanzarme sin miedo a aquello que me apasiona, gracias.

Resumen

Este documento se centra en la investigación y el estudio de modelos discriminativos de voces generadas artificialmente, de forma que se pueda concluir si la parte compleja de la señal de la voz es una característica importante a la hora de clasificar la misma.

En el documento, primero se hablará de conceptos clave como son los tipos de spoofing y de generación de voz más comunes actualmente; y se revisarán las nociones base de la creación de los espectrogramas que representan la señal de voz, es decir, la transformada de Fourier y los algoritmos optimizados, la Transformada Rápida de Fourier (FFT) y la Transformada de Fourier de Tiempo Reducido (STFT) y, seguidamente, se llevará acabo la parte práctica realizando distintos experimentos al introducir las nuevas características de la parte compleja en el modelo.
Finalmente, se analizarán los resultados obtenidos en los distintos experimentos para obtener así las conclusiones.

Palabras clave:

- Transformada de Fourier
- FFT
- STFT
- Deepfakes
- Detección de spoofing
- Clasificación de voces

Abstract

This paper focuses on the investigation and study of discriminative models of artificially generated speech, so as to conclude whether the complex part of the speech signal is an important feature when classifying speech.

In the paper, we will first discuss key concepts such as the most common types of spoofing and speech generation nowadays; and we will review the basic notions of creating spectrograms representing the speech signal, and the optimized algorithms, the Fourier Transform (FFT) and the Reduced Time Fourier Transform (STFT), and then the practical part will be carried out by by performing different experiments by introducing the new features of the complex part into the model. Finally, the results obtained in the different experiments are analyzed in order to draw conclusions.

Palabras clave:

- Fourier Transform
- FFT
- STFT
- Deepfakes
- Spoofing detection
- Voice classification

Índice de contenidos

Índice de tablas **XIII**

Índice de figuras **XVII**

1. Introducción **1**
1.1. Contexto y alcance 1

2. Objetivos **3**
2.1. Modelización de la parte compleja de la serie de Fourier generada por voz 3
2.2. Estudio de las características sobre el modelo discriminativo . . . 4

3. Metodología **5**
3.1. Series de Fourier 6
3.1.1. Modelización de las series de Fourier 6
3.1.2. Transformada de Fourier 9
3.1.3. Aplicación de la Transformada de Fourier 10
3.2. Estudio de las características sobre el modelo discriminativo . . . 15
3.2.1. HABLA 15
3.2.2. Tratamiento de los datos 15
3.2.3. Espectrogramas y estudio de las características 16
3.2.4. Filtros MEL y MFCC 19
3.2.5. Características 19

4. Resultados **24**
4.1. Diagramas para el estudio del desempeño del modelo 25
4.1.1. Matriz de confusión 25
4.1.2. Diagrama de la relevancia de las características 27
4.2. Experimento base 28
4.3. Experimento con las características de la parte imaginaria 30
4.4. Experimento con las características de la parte imaginaria y la real 36
4.5. Experimento con las características de la parte real 41
4.6. Comparación de resultados 45

5. Conclusiones **47**

6. Trabajos futuros **50**

Bibliografía **51**

Apéndices **55**

A. Gráficas individuales del desempeño de características **57**
A.1. Características del primer experimento 57
A.2. Características del segundo experimento 60
A.2.1. Características de la parte compleja 60
A.2.2. Características de la parte real 63
A.3. Características del tercer experimento 66

Índice de tablas

4.1. Métricas de rendimiento del experimento base. 28
4.2. Métricas de rendimiento del primer experimento. 30
4.3. Métricas de rendimiento del segundo experimento. 36
4.4. Métricas de rendimiento del tercer experimento. 41
4.5. Comparación de las métricas de rendimiento entre los diferentes experimentos. 45

Índice de figuras

3.1. Diagrama de la Transformada de Fourier de Tiempo Reducido. . . 14
3.2. Diagrama de la clasificación de muestras en HABLA. 15
3.3. Gráficas temporales inicialmente de los audios “arf_00295_01212219131” y “arf_00295_01212219131_TTS”, respectivamente. 16
3.4. Gráficas temporales tras eliminar los silencios de los audios “arf_00295_01212219131” y “arf_00295_01212219131_TTS”, respectivamente. 17
3.5. Espectrogramas de la parte real y la parte compleja de la señal “arf_00295_01212219131”. 18
3.6. Espectrogramas de la parte real y la parte compleja de la señal “arf_00295_01212219131_TTS”. 18
3.7. Histograma de la parte imaginaria de de la señal “arf_00295_01212219131”. 20
3.8. Gráfico Q-Q de la parte imaginaria de de la señal “arf_00295_01212219131”. 20
3.9. Resultado de las prueba de Kolmogórov-Smirnov sobre la parte compleja de la señal “arf_00295_01212219131”. 20
3.10. Cálculo de las características sobre una muestra del Dataset. . . . 22

4.1. Distribución de las muestras dentro del conjunto de entrenamiento. 25
4.2. Matriz de confusión del experimento base. 29
4.3. Ranking de características del experimento base. 30
4.4. Matriz de confusión del primer experimento. 31
4.5. Ranking de características del primer experimento. 32
4.6. Diagrama de desempeño de la característica “ImagPartStd” en el primer experimento. 32
4.7. Histograma de la distribución de la característica “ImagPartStd”. 33
4.8. Diagrama de densidad de la distribución de la característica “ImagPartStd”. 33
4.9. Diagrama de desempeño de la característica “ImagPartMean” en el primer experimento. 34
4.10. Histograma de la distribución de la característica “ImagPartMean”. 35
4.11. Diagrama de densidad de la distribución de la característica “ImagPartMean”. 35
4.12. Matriz de confusión del segundo experimento. 36
4.13. Ranking de características del segundo experimento. 37

4.14. Diagrama de desempeño de la característica “RealPartMean” en el segundo experimento. 38
4.15. Histograma de la distribución de la característica “RealPartMean”. 38
4.16. Diagrama de densidad de la distribución de la característica “RealPartMean”. 39
4.17. Histograma de la distribución de la característica “RealPartMean” para los valores “bonafide”. 39
4.18. Diagrama de desempeño de la característica “ImagPartMean” en el segundo experimento. 40
4.19. Diagrama de desempeño de la característica “ImagPartStd” en el segundo experimento. 40
4.20. Matriz de confusión del tercer experimento. 41
4.21. Ranking de características del tercer experimento. 42
4.22. Diagrama de desempeño de la característica “RealPartMean” en el tercer experimento. 43
4.23. Diagrama de desempeño de la característica “RealPartStd” en el tercer experimento. 43
4.24. Histograma de la distribución de la característica “RealPartStd”. 44
4.25. Diagrama de densidad de la distribución de la característica “RealPartStd”. 44

A.1. Diagrama de desempeño de la característica “ImagPartStd” en el primer experimento. 58
A.2. Diagrama de desempeño de la característica “ImagPartMean” en el primer experimento. 58
A.3. Diagrama de desempeño de la característica “ImagPartPercentile_5” en el primer experimento. 58
A.4. Diagrama de desempeño de la característica “ImagPartPercentile_10” en el primer experimento. 59
A.5. Diagrama de desempeño de la característica “ImagPartPercentile_50” en el primer experimento. 59
A.6. Diagrama de desempeño de la característica “ImagPartPercentile_90” en el primer experimento. 59
A.7. Diagrama de desempeño de la característica “ImagPartPercentile_95” en el primer experimento. 60
A.8. Diagrama de desempeño de la característica “ImagPartMean” en el segundo experimento. 60
A.9. Diagrama de desempeño de la característica “ImagPartStd” en el segundo experimento. 61
A.10. Diagrama de desempeño de la característica “ImagPartPercentile_5” en el segundo experimento. 61
A.11. Diagrama de desempeño de la característica “ImagPartPercentile_10” en el segundo experimento. 61

A.12.Diagrama de desempeño de la característica “ImagPartPercentile_50” en el segundo experimento. 62
A.13.Diagrama de desempeño de la característica “ImagPartPercentile_90” en el segundo experimento. 62
A.14.Diagrama de desempeño de la característica “ImagPartPercentile_95” en el segundo experimento. 62
A.15.Diagrama de desempeño de la característica “RealPartMean” en el segundo experimento. 63
A.16.Diagrama de desempeño de la característica “RealPartStd” en el segundo experimento. 63
A.17.Diagrama de desempeño de la característica “RealPartPercentile_5” en el segundo experimento. 64
A.18.Diagrama de desempeño de la característica “RealPartPercentile_10” en el segundo experimento. 64
A.19.Diagrama de desempeño de la característica “RealPartPercentile_50” en el segundo experimento. 64
A.20.Diagrama de desempeño de la característica “RealPartPercentile_90” en el segundo experimento. 65
A.21.Diagrama de desempeño de la característica “RealPartPercentile_95” en el segundo experimento. 65
A.22.Diagrama de desempeño de la característica “RealPartMean” en el tercer experimento. 66
A.23.Diagrama de desempeño de la característica “RealPartStd” en el tercer experimento. 66
A.24.Diagrama de desempeño de la característica “RealPartPercentile_5” en el tercer experimento. 67
A.25.Diagrama de desempeño de la característica “RealPartPercentile_10” en el tercer experimento. 67
A.26.Diagrama de desempeño de la característica “RealPartPercentile_50” en el tercer experimento. 67
A.27.Diagrama de desempeño de la característica “RealPartPercentile_90” en el tercer experimento. 68
A.28.Diagrama de desempeño de la característica “RealPartPercentile_95” en el tercer experimento. 68

1 Introducción

1.1. Contexto y alcance

Hablamos de *deepfakes* refiriéndonos al uso de métodos de deep learning para la alteración importante de las características de un audio o video, o para directamente la creación de éstos. La generación artificial de contenido audiovisual realista es quizá uno de los avances de los últimos años con un impacto social más perceptible. Comenzando por imágenes ante las que un ojo no experto no es capaz de determinar si esa fotografía fue realmente tomada en algún momento o si ha sido generada por una máquina; hoy en día, la tecnología ha alcanzado ese mismo punto en cuando a vídeos o contenido de audio. Evidentemente es un área muy importante de investigación pues son innumerables los beneficios que hay en ella. En concreto, cuando se habla de generación artificial de voces, se presentan los siguientes intereses entre otros:

- **Asistencia en el aprendizaje de lectura.**

  Es mucho más útil que puedan contar con una herramienta que les permita practicar cuantas veces necesiten y que esta sea tan precisa como una persona real, ya no sólo en pronunciación, sino también en factores muy importantes como la pronunciación o la emotividad que se le aporta al mensaje.

- **Traducción de contenido y mensajes.**

  Por un lado, los creadores de contenido y empresas podrán alcanzar de forma cómoda a un mayor público sin la limitación del lenguaje. Por otro

lado, esto se podría aplicar a que las aplicaciones de traducción simultánea que ya existen se vean perfeccionadas y favorezca una comunicación más natural entre hablantes de distintas lenguas.

- **Apoyo a personas con dificultades de habla.**

  En el caso de personas que por ciertas circunstancias hayan perdido parcial o totalmente la facilidad para hablar, se podría generar una herramienta que ponga voz a lo que quieren decir, su voz si se cuenta con un registro previo de ellos hablando. De esta forma tendrían más naturalidad al relacionarse y comunicarse mediante su propia voz y el impacto antes las circunstancias de la pérdida sería menor. En líneas de investigación similares se está trabajando con pacientes de ELA, para que, una vez llegue el punto de la enfermedad donde se les paralizan las cuerdas vocales, puedan seguir usando su voz aunque sea generada por una máquina.

Sin embargo, no todo es positivo en que se alcance un punto donde la voz humana sea indistinguible de una generada por una máquina. Se plantean unas circunstancias donde es casi imposible asegurar los pilares de integridad y autenticidad en cuanto a seguridad informática se debe. No tener buenas formas de discernir la realidad por medios digitales supone un gran problema en el momento en el que los personas malintencionadas tienen la posibilidad de utilizar la tecnología para suplantar la identidad, entre otro tipo de ataques o fraudes.

Existen principalmente cuatro tipos de ataques de voz falsa: suplantación mediante voice impersonation attack, conversión de voz (usando un audio de base para transformarlo a la voz deseada manteniendo el mensaje), síntesis de texto a voz (TTS) y repetición (cuando se realizan grabaciones de una voz real para usarla posteriormente a voluntad del atacante).[1] Las tres últimas técnicas son tecnologías de relativamente fácil acceso, por lo que la alarma respecto a los peligros que se presentan está completamente fundamentada.
Nadie está libre de estas nuevas vulnerabilidades que se han presentado junto al perfeccionamiento de estas tecnologías y es de vital importancia conseguir herramientas que permitan su detección para minimizar los daños.

Además, teniendo en cuenta que son tecnologías en auge y cada vez más accesibles, la curva en la que aumenta la cantidad de este tipo de modelos es exponencial y deja de reducirse a únicamente los modelos que se generan en entornos de investigación profesional o educativa. Cada día se avanza en este campo y surgen evoluciones y combinaciones de estas tecnologías, en este mismo proyecto con voces generadas por uno modelo que une mode TTS y Difusión.
Se puede asegurar que habrá modelos existentes de los que no la gran mayoría de la población no tiene conocimiento aún y para los que es fundamental seguir desarrollando e investigando en la parte de la detección de los $deepfakes$.

2

Objetivos

El objetivo principal de este Trabajo Fin de Grado es la mejora en los modelos discriminativos que se encargan de la detección de *deepfakes* de de voz usados para ataques de spoofing. Un ataque de spoofing es aquel en el que realizan una suplantación de identidad con fines fraudulentos, en este caso, replicando la voz de, o bien la víctima, si pretenden por ejemplo suplantarle la identidad ante entidades oficiales como puede ser el banco, o bien un conocido de la víctima, para que sea ella misma quien les proporcione la información pensando que está ayudando a su familiar o amigo.

En concreto, se plantea como hipótesis de estudio que la parte compleja de la señal de audio o de voz que se ha de clasificar supone realmente una característica relevante al momento de la clasificación realizada por el modelo. Por tanto el trabajo va a tener dos partes principales y claramente distinguibles:

2.1. Modelización de la parte compleja de la serie de Fourier generada por voz

Con las series de Fourier se realiza una modelización física de las voces en forma de espectrogramas. Es en la creación de estos espectrogramas donde usualmente se desecha la parte compleja y se usa únicamente la parte real para su posterior introducción en los modelos o redes neuronales.

En este proyecto el objetivo es generar los espectrogramas de ambas parte real

y compleja, de forma que luego se pueda considerar la parte compleja como un añadido al análisis sobre los espectrogramas de la parte real. Para dicho cometido se estudiará la base teórica de la transformada de Fourier y, evidentemente, su forma compleja, además de los algoritmos que se usan para su cálculo y posterior procesamiento.

2.2. Estudio de las características sobre el modelo discriminativo

Haciendo uso del dataset HABLA[2], se pretende comparar el resultado que se obtendría de aplicar un modelo para la clasificación binaria sobre las voces con las características usuales con el resultado que proporcionaría si se incluyen además algunas características calculadas sobre la parte compleja de la voz.
El modelo será de glassbox y se entrenará de forma que sea interpretable haciendo uso de InterpretML[3]. Con el fin de completar satisfactoriamente este apartado, se tendrán los siguientes subobjetivos:

1. Selección y estudio de distintas propiedades de la parte compleja de la señal de voz.
2. Comparación de las características calculadas sobre la parte compleja junto con aquellas características que suelen usarse para la clasificación y análisis de voces.
3. Identificación de las características más relevantes para la clasificación de voces reales y falsa.
4. Estudio del desempeño global del modelo en los distintos experimentos.

3

Metodología

Durante la realización de este proyecto se han utilizado principalmente las siguientes herramientas informáticas:

- **HABLA:**[2] Dataset en español compuesto de voces reales y falsas, generadas por distintas técnicas de *deepfakes*, que sirve de base para el experimento.
- **Google Colab**[4] usado para albergar el código base para la configuración y ejecución del experimento. Se hace uso de Python para trabajar con la gran cantidad de librerías que existen enfocadas a Machine Learning. En concreto, se puede destacar el uso de *librosa*[5] para la creación de espectrogramas y la carga de audios o *pandas* para el uso y tratamiento de dataframes.
- **Explainable Boosting Machine:**[6] Modelo de aprendizaje automático que se usa para el entrenamiento del experimento. Es de tipo glassbox, lo que proporciona interpretabilidad, pero manteniendo un rendimiento similar al de Random Forest. Pertene a los Generalized Additive Models (GAMs), cuyas predicciones Y se calculan como $Y = \beta_0 + f(X_1) + f(X_2) + f(X_3) + \ldots + f(X_n)$, donde las funciones sobre cada variable también se calculan mediante técnicas de Machine Learning, lo que consigue el buen rendimiento del modelo.[7]
- **InterpretML:**[3] toolkit para interpretar distintos modelos, tanto glassbox como blackbox. En el proyecto se utiliza junto al EMB para conseguir una mejor interpretabilidad que si se aplicara a modelos blackbox.

- **OpenSmile:**[8] herramienta para extraer las características del audio de las voces. En concreto, se usa el dataset de GeMaps.

3.1. Series de Fourier

Las series de Fourier permiten descomponer la voz en sus componentes frecuenciales, de esta forma se consiguen diferenciar sus características principales, tales como serían las notas de una melodía, a la vez que se podrán detectar algunos sutiles detalles. A través de dicha descomposición de los factores frecuenciales, se logrará detectar patrones que permitirán la clasificación de voces reales y generadas. Es por esto que es una herramienta útil y eficaz para el problema que aquí se trata. Representan las señales de forma precisa y eficiente para su posterior análisis, asegurando al aplicarlas un resultado sólido.

Las series de Fourier representan funciones periódicas como combinación lineal de funciones sinusoidales y cosenoidales con el mismo período T, es decir, una serie temporal trigonométrica.

3.1.1. Modelización de las series de Fourier

Una serie temporal o cronológica es una sucesión de datos medidos en determinados momentos y ordenados cronológicamente, prestando especial atención al orden ya que no son independientes entre sí y uno depende de los datos medidos previamente. Aunque el período de medición de los datos puede ser el mismo o diferente, como ya se ha indicado, para el análisis de las voces se trabajará con un período constante.

Estas series temporales se pueden representar mediante una serie de Fourier, pero para encontrar la serie exacta que la representa, se ha de calcular los llamados Coeficientes de Fourier.
De forma genérica la serie de Fourier se plantea en este caso como una serie trigonométrica de período 2π:

$$\frac{a_0}{2} + \sum_{k=1}^{\infty}(a_k cos kx + b_k sen kx) \tag{3.1}$$

Suponiendo que la serie planteada converge en el período $[-\pi, \pi]$ a una función f, se tiene la igualdad:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_k cos kx + b_k sen kx) \tag{3.2}$$

Los coeficientes a encontrar son a_k y b_k.

Dado que se va a trabajar en un entorno computacional binario, se estudia un algoritmo presentado por Cooley-Tukey para el cálculo de los coeficientes. Es un requisito que la función que establece la serie trigonométrica se plantee de forma de compleja.

Para conseguir representar la función en forma compleja, primero se recuerda la fórmula de Euler:

$$e^{ix} = cosx + isenx \tag{3.3}$$

de la cual se obtiene, tras sumar y restar respectivamente e^{ix} y e^{-ix}, las dos igualdades referentes al seno y coseno:

$$cost = \frac{e^{ix} + e^{-ix}}{2} \tag{3.4}$$

$$sent = \frac{e^{ix} - e^{-ix}}{2i} \tag{3.5}$$

Introduciendo estas igualdades en la función (3.2),

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_k\frac{e^{ikx} + e^{-ikx}}{2} + b_k\frac{e^{ikx} - e^{-ikx}}{2i})$$

Y limpiando para evitar la división de complejos,

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_k\frac{e^{ikx} + e^{-ikx}}{2} - ib_k\frac{e^{ikx} - e^{-ikx}}{2})$$

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(\frac{1}{2}(a_k(e^{ikx} + e^{-ikx}) - ib_k(e^{ikx} - e^{-ikx}))$$

Sacando factor común,

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(\frac{1}{2}((a_k - ib_k)(e^{ikx}) + (a_k + ib_k)(e^{-ikx}))) \tag{3.6}$$

Considerando que los coeficientes de Fourier de la transformada en reales son de la forma[9]:

$$a_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)coskxdx$$

$$b_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)senkxdx$$

Se plantean en forma compleja:

$$a_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)[\frac{e^{ikx}+e^{-ikx}}{2}]dx$$

$$b_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)[\frac{e^{ikx}-e^{-ikx}}{2i}]dx$$

Ahora se procede a buscar la forma de la fórmula (3.6) en función de los coeficientes complejos recién planteados.

$$\begin{aligned} a_k - ib_k =& \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)[\frac{e^{ikx}+e^{-ikx}}{2}]dx - i(\frac{1}{\pi}\int_{-\pi}^{\pi} f(x)[\frac{e^{ikx}-e^{-ikx}}{2i}]dx) \\ =& \frac{1}{2\pi}(\int_{-\pi}^{\pi} f(x)2e^{-ikx}dx) = \frac{1}{\pi}(\int_{-\pi}^{\pi} f(x)e^{-ikx}dx) = A_k \end{aligned}$$

$$\begin{aligned} a_k + ib_k =& \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)[\frac{e^{ikx}+e^{-ikx}}{2}]dx + i(\frac{1}{\pi}\int_{-\pi}^{\pi} f(x)[\frac{e^{ikx}-e^{-ikx}}{2i}]dx) \\ =& \frac{1}{2\pi}(\int_{-\pi}^{\pi} f(x)2e^{ikx}dx) = \frac{1}{\pi}(\int_{-\pi}^{\pi} f(x)e^{ikx}dx) = B_k \end{aligned}$$

Introduciendo estos nuevos coeficientes,

$$\begin{aligned} f(x) =& \frac{a_0}{2} + \sum_{k=1}^{\infty}(\frac{1}{2}(A_k(e^{ikx}) + B_k(e^{-ikx}))) \\ =& \frac{a_0}{2} + \sum_{k=1}^{\infty}(A'_k(e^{ikx}) + B'_k(e^{-ikx})) \end{aligned}$$

Por facilidad de lectura, se denota A' como A al ser coeficientes que difieren en contantes. De igual forma, de ahora en adelante se obviará el cambio y directamente se tratarán los coeficientes con la letra principal, sin reflejar el cambio. Introduciendo el término a_0 en el sumatorio,

$$\begin{aligned} f(x) =& \frac{a_0}{2} + \sum_{k=1}^{\infty}(A_k(e^{ikx}) + B_k(e^{-ikx})) \\ =& \sum_{k=0}^{\infty} A_k(e^{ikx}) + \sum_{k=1}^{\infty} B_k(e^{-ikx}) \end{aligned}$$

Se puede observar que

$$A_k = \frac{1}{\pi}(\int_{-\pi}^{\pi} f(x)e^{-ikx}dx) = B_{(-k)}$$

Por lo que se puede aplicar esta igualdad en la fórmula invirtiendo el sumatorio.

$$\begin{aligned} f(x) =& \sum_{k=0}^{\infty} A_k(e^{ikx}) + \sum_{k=-\infty}^{-1} B_{(-k)}(e^{-ikx}) \\ =& \sum_{k=0}^{\infty} A_k(e^{ikx}) + \sum_{k=-\infty}^{-1} A_k(e^{ikx}) \end{aligned}$$

$$f(x) = \sum_{k=-\infty}^{\infty} A_k(e^{ikx}) \tag{3.7}$$

Obteniendo finalmente así la forma compleja de las series de Fourier.[10]

3.1.2. Transformada de Fourier

A partir de la forma compleja se puede hallar fácilmente la transformada de Fourier, que es la que va a permitir representar funciones no periódicas y toda la recta, que son precisamente las que nos interesan al trabajar con ondas de sonido.

La función ha de ser periódica con período 2L, integrable y definida en todo $\mathbb{R}$. Usando la expresión (3.7) e introduciendo los cambios de parámetros correspondientes mediante un cambio de variable $t = x\frac{L}{\pi}$:

$$f(t) = \sum_{k=-\infty}^{\infty} A_k(e^{ik\pi t/L}) = \sum_{k=-\infty}^{\infty} \frac{1}{2L}(\int_{-L}^{L} f(t)e^{-ikt\pi/L}dt)(e^{ikt\pi/L})$$

Ahora denotando por:

$$\begin{aligned} \xi_k =& \frac{k}{2L} \\ g(\xi) =& \int_{-L}^{L} f(t)e^{-2\pi it\xi}dt \end{aligned}$$

De forma que la función se puede expresar como:

$$\sum_{k=-\infty}^{\infty} (\xi_k - \xi_{k-1})g(\xi_k)e^{2\pi i\xi_k x}$$

Se puede ver que tiene la forma de las sumas de Riemman, por lo que al tomar límites se obtiene la fórmula de inversión que da f a partir de la transformada de Fourier.

$$f(t) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(t) e^{-2\pi i \xi t} dt \right] e^{2\pi i \xi t} d\xi. \tag{3.8}$$

Siendo finalmente la transformada de Fourier:

$$\int_{k=-\infty}^{\infty} f(t) e^{-2\pi i t \xi} dt \tag{3.9}$$

3.1.3. Aplicación de la Transformada de Fourier

Cuando se quiere aplicar la transformada de Fourier surgen principalmente dos problemas.
El primero es que el input que se va a tener no es una función analítica sino una serie discreta, ya que los ordenadores no trabajan realmente con entornos continuos y al grabar la voz, lo que se consigue es registrar el audio como una serie de puntos. De igual manera, el cálculo de las integrales se hace numéricamente, lo que quiere decir que será aproximado. El segundo problema con el que uno se va a encontrar es que, sin restricciones ni condiciones especiales, el cálculo de la transformada es bastante costoso tanto de recursos como de tiempo.

Transformada Discreta de Fourier

Para solucionar el primero de los problemas sean funciones definidas en $\mathbb{N} \cup \{0\} \to \mathbb{C}$. Suponer que la señal recibida tiene módulo N, entonces es una sucesión periódica de período N, tal que $f[k+N] = f[k]\ \forall k$. Llevando esto a la transformada de Fourier se tiene que a cada función la va a transformar de la siguiente manera:

$$f'[k] = \sum_{j=0}^{N-1} f[j] e^{-2\pi j k / N} \tag{3.10}$$

Siendo de nuevo la función $f'[k]$ periódica de período N.

El cálculo de esta transformada, la transformada discreta de Fourier (DFT) tiene una complejidad del orden $0(N^2)$. Sin embargo, si se consideran únicamente los casos con N par, se reduce la complejidad al orden $O(NlogN)$. Para asegurar que se cumpla esta condición, considérese lo siguiente:
Sea $\omega = e^{-2\pi j/N}$ y $\omega^{-1} = e^{2\pi j/N}$, entonces se tiene que $w^{jN} = 1\ \forall j \in \mathbb{N}$;

y $1+\omega^j+\omega^{2j}+...+\omega^{(N-1)j}$ es N cuando j es múltiplo de N y 0 en cualquier otro caso.
Retomando la transformada 3.10,

$$f'[k] = \sum_{j=0}^{N-1} f[j]e^{-2\pi jk/N} = \sum_{j=0}^{N-1} f[j]\omega^k \equiv f' = A(\omega)f \tag{3.11}$$

donde

$$A(\omega) = \begin{pmatrix} 1 & 1 & 1 & ... & 1 \\ 1 & \omega & \omega^2 & ... & \omega^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & ... & \omega^{(N-1)(N-1)} \end{pmatrix} \tag{3.12}$$

Transformada rápida de Fourier

Como se dijo anteriormente, dadas estas condiciones, existe un algoritmo para el cálculo de la transformada que reduce la complejidad $O(NlogN)$. Este algoritmo fue desarrollado por Cooley y Tukey en 1965 [11].

Está basado en la estrategia de "Divide y Vencerás", que se encarga de dividir el problema en subproblemas más sencillos que se resolverán de forma recursiva. A continuación se presenta el pseudocódigo del mismo:

Algoritmo 1 *Transformada Rápida de Fourier (FFT)*

1: Reordenar los elementos de la matriz $A(\omega)$ de tal forma que queden primero las filas correspondientes a N par.
2: Análogamente, reordenar los elementos considerando esta vez las columnas para que queden primero las correspondientes a N par.
3: La matriz resultante será de la forma:

$$A(\omega) = \begin{pmatrix} 1 & 1 & ... & 1 & 1 & 1 & ... & 1 \\ 1 & \omega^2 & ... & \omega^{N-2} & \omega & \omega^3 & ... & \omega^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{N-2} & ... & \omega^{(\frac{N}{2}-1)^2} & \omega^{(\frac{N}{2}-1)} & \omega^{3(\frac{N}{2}-1)} & ... & \omega^{(\frac{N}{2}-1)(\frac{N}{2})} \\ 1 & 1 & ... & 1 & -1 & -1 & ... & -1 \\ 1 & \omega^2 & ... & \omega^{N-2} & -\omega & -\omega^3 & ... & -\omega^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \omega^{N-2} & ... & \omega^{(\frac{N}{2}-1)^2} & -\omega^{(\frac{N}{2}-1)} & -\omega^{3(\frac{N}{2}-1)} & ... & -\omega^{(\frac{N}{2}-1)(\frac{N}{2})} \end{pmatrix}$$

4: Vemos que la matriz está compuesta por cuatro submatrices de dimensión $\frac{N}{2} \times \frac{N}{2}$ que cumplen

$$A(\omega) = \begin{pmatrix} B & C \\ B & -C \end{pmatrix} \; con \; B \in \mathbb{C}^{\frac{N}{2}}$$

5: y donde $f_p[j] = f[2j]$ y $f_i[j] = f[2j+1] \; \forall j = 0, ..., (\frac{N}{2} - 1)$
6: Iterar hasta que $B \in \mathbb{C}^2$

Sin embargo, la Transformada Rápida de Fourier homogeiniza el valor de la frecuencia en toda la longitud del audio, por lo que se pierde una gran cantidad de detalle. Es necesaria la medición de las frecuencias a lo largo del tiempo para el procesamiento de la voz, para lo cual se hace uso de la Transformada de Fourier de Tiempo Reducido (Short Term Fourier Transform, STFT abreviando). Esta herramienta está íntimamente ligada a la FFT puesto que se encarga de calcularla en segmentos de la señal, lo que se llaman *ventanas*.

Transformada de Fourier de Tiempo Reducido

Cuando una señal de audio se registra digitalmente se puede considerar como un array de puntos que representan o grafican la misma. Con la STFT se elige una ventana de dimensión n y un salto de tamaño w, de esta forma, se cogerán los primeros n puntos de la señal y se calculará la FFT sobre ellos. Seguidamente

se salta el número de puntos que indique w y se vuelve a seleccionar una ventana de n puntos. Repitiendo este proceso, se alcanza el final de la señal de audio.
Evidentemente, dependiendo del tamaño n y w se obtendrán unos resultados u otros, al punto de que si $n < w$ quedarían puntos sin cubrir, y a medida que la n aumenta respecto a la w, aumenta el solapamiento y más puntos se comparten entre ventanas colindantes. Esto influye en el rendimiento y dependiendo de la aplicación de la STFT es más apropiado usar unos tamaños u otros. La longitud de la ventana en la STFT está relacionada con la resolución de la señal. Las ventanas largas proporcionan una mayor resolución de frecuencia pero una menor resolución temporal, mientras que las ventanas cortas proporcionan una mayor resolución temporal pero una menor resolución de frecuencia.
Por construcción de la herramienta, es necesario que el tamaño de la ventana sea potencia de 2 (generalmente se seleccionan valores entre 16 y 2048) y se suelen considerar solapamientos del 0 %, 25 %, 50 %, 75 %. Dependiendo de la cantidad de solapamiento, se puede llegar a perder más información, por lo que se recomienda un mayor solapamiento a medida que aumenta el tamaño de la ventana.

Para el experimento se ha considerado una ventana de $n = 512$ y un solapamiento del 50 %, es decir, un salto de $w = 256$. Los audios del dataset tienen una frecuencia de muestreo de 16.000 Hz, es decir que se recorren 16.000 muestras por segundo, así que por regla de tres se sabe que la ventana tendrá una duración, lo que es el tamaño en el espacio de tiempo, de 32 milisegundos. Se seleccionó ese tamaño de ventana por ser relativamente intermedio, proporcionando unos niveles de detalle tanto para frecuencias como para la variable temporal bastante balanceados, ya que de media se capturan unos tres o cuatro fonemas en este tipo de ventanas. Por cómo se usan las voces en el proyecto, no existe una prioridad clara sobre una u otra variable y se consideró lo más apropiado intentar alcanzar el máximo de detalle en ambas variables sin perjudicar a ninguna. Consecuentemente, el tamaño del salto se elige en función del solapamiento deseado, y éste está recomendado por estudios previos. [12]

Considerando este tamaño de ventana y salto, y sabiendo que la cantidad aproximada de puntos de las señales de audio de este dataset rondan entre los 90.000 y 100.000 puntos y que al normalizarlas se quedan entre los 30.000 y 40.000, se pueden calcular el número de ventanas que se crearán para cubrir toda la señal y, de igual forma, las dimensiones de la matriz compleja resultante de la operación.

Por ejemplo, en el caso del audio "arf_00295_01212219131" de la base de datos HABLA se tiene que una vez normalizado es un array de 31.744 puntos, por lo que si los cuadramos en ventanas de 512 puntos con el salto de 256, teniendo la primera ventana que abarca 512 puntos, pero la siguiente irá a la mitad de esta por lo que abarcará sólo 256 puntos. Entonces si se dividen los 31.744 entre 256 que son realmente el tamaño de las fracciones que saldrán, se obtienen un total de 124 ventanas. La matriz tendrá dimensión, entonces, (257, 125), ya que habrá

uno de más por los extremos donde no se encuentra solapamiento.
Si se observa el diagrama de la figura 3.1, se puede ver que al realizar el proceso de ventanado existen cuatro ventanas (o crestas) mientras que en la línea temporal se refleja como cinco segmentos de la mitad de longitud de lo que abarca la ventana.

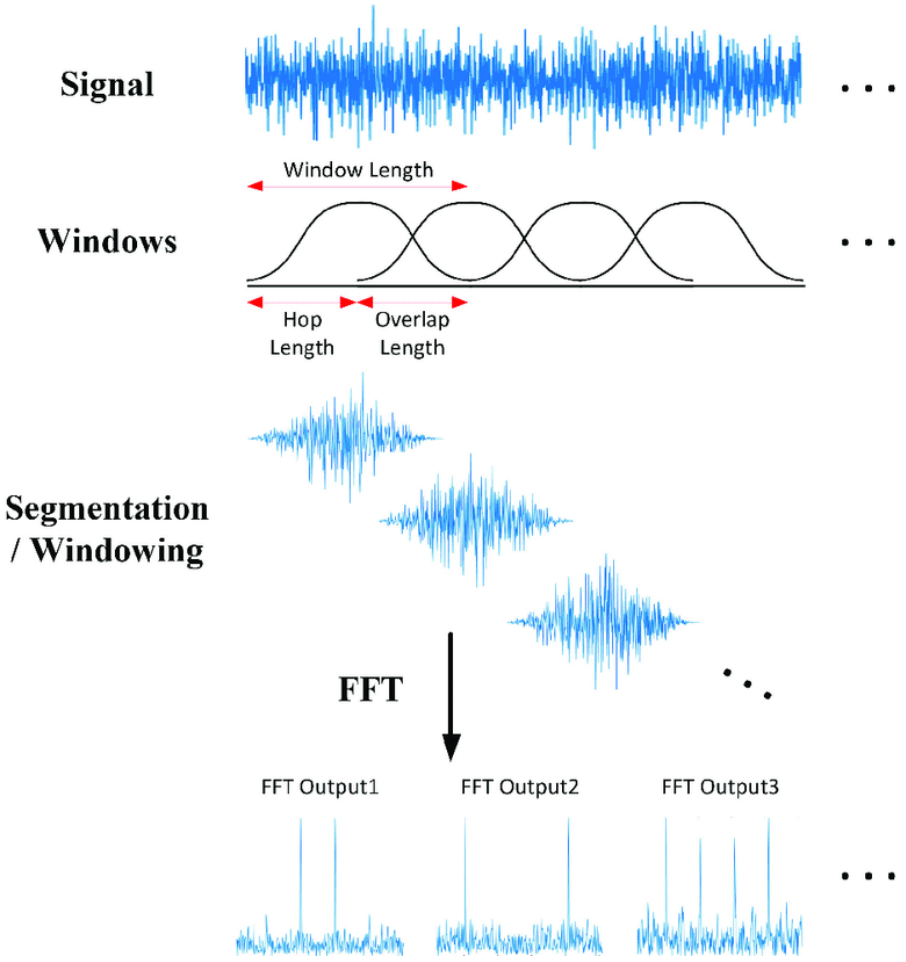


Figura 3.1: Diagrama de la Transformada de Fourier de Tiempo Reducido. [12]

3.2. Estudio de las características sobre el modelo discriminativo

3.2.1. HABLA

Para el proyecto se va a trabajar con el dataset HABLA, un conjunto de grabaciones de voz en español latinoamericano especialmente generado para el estudio, entrenamiento y validación de modelos anti-spoofing de voz. [2]

El dataset se compone de casi 23.000 grabaciones reales y 58.000 grabaciones generadas mediante diversas técnicas de *deepfakes*: CycleGAN [13], modelos de Difusión [14], StarGan [15] y TTS (tanto únicamente TTS, como Diff-TTS y TTS-StarGAN) [16].
En la figura 3.2 se puede observar la distribución por clases de los ficheros de audio de HABLA en función de la cantidad de ficheros que contiene cada una, siendo la sección azul la correspondiente a los audios reales llamados "bonafide".

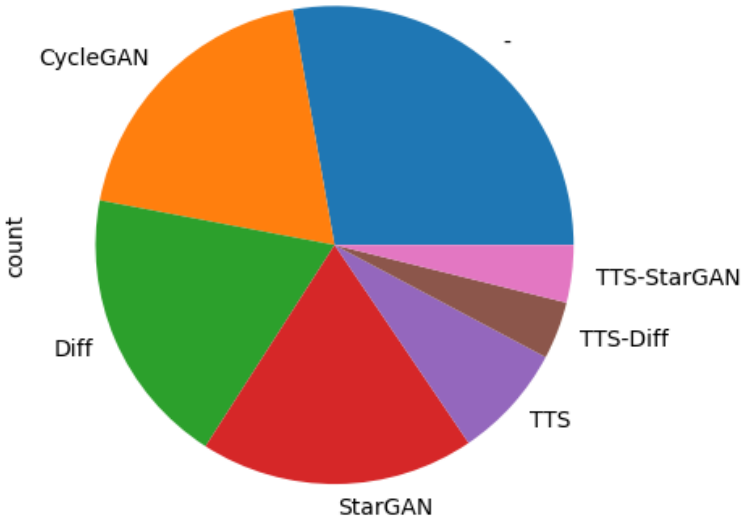


Figura 3.2: Diagrama de la clasificación de muestras en HABLA.

3.2.2. Tratamiento de los datos

El primer paso al trabajar con datos siempre es normalizarlos y tratarlos. En este caso, los datos se encuentran muy bien preparados, e incluso en el propio nombre de cada fichero de voz se puede observar si es audio real o generado y

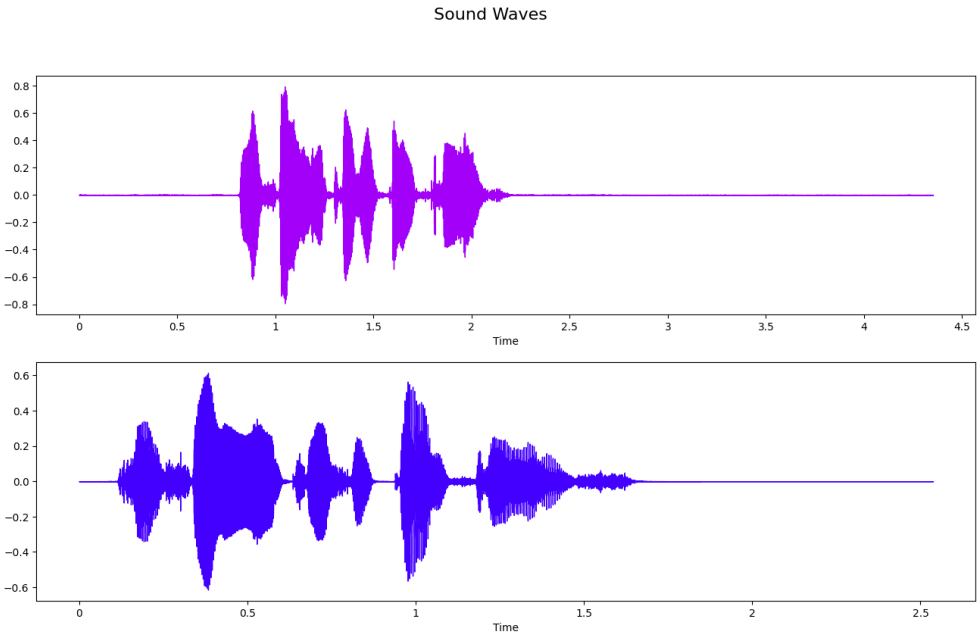


Figura 3.3: Gráficas temporales inicialmente de los audios "arf_00295_01212219131" y "arf_00295_01212219131_TTS", respectivamente.

la demografía a la que pertenece el hablante: acento y género. Por ejemplo, considérese el fichero "StarGAN-vem_07508_01534566632-vem_09697_0185458", entonces se ve que es un deepfake generado mediante StarGAN simulando una voz masculina con acento de Venezuela (StarGAN-vem).

Para trabajar en la siguiente sección con los espectrogramas evitando diferencias que no se deban a la voz, se realizó una normalización adicional de los audios eliminando los silencios que rodean la voz, de forma que audios como los de la figura 3.3 pasaron a ser como los que se encuentran en la figura 3.4.

3.2.3. Espectrogramas y estudio de las características

Un espectrograma es una imagen que representa el espectro de frecuencias de una señal a medida que varía con el tiempo y a través de él podemos ver los niveles de la intensidad de la señal o sonoridad (generalmente medidos los niveles de energía en dB). Se pueden ver los espectrogramas como un mapa de calor, es decir, como una gráfica en la que la intensidad se muestra variando el color o el brillo y donde los ejes son el tiempo y la frecuencia.

Como se vio en la sección 3.1.3, se van a generar mediante la STFT. La STFT se calculará con la librería *librosa*, que está enfocada al tratamiento de audio y tiene tanto funciones para el procesamiento de estos, como aquellas relacionadas

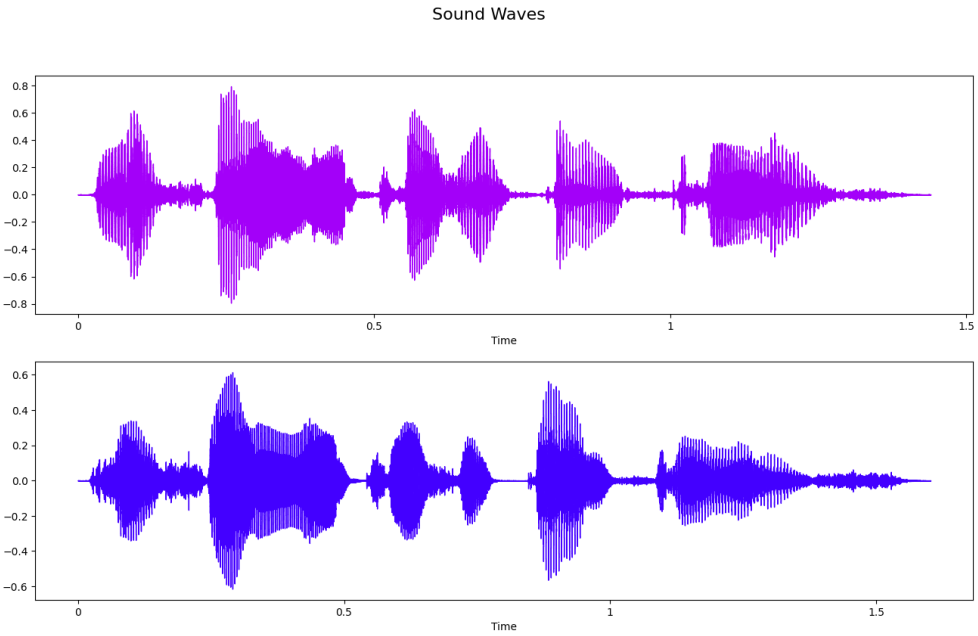


Figura 3.4: Gráficas temporales tras eliminar los silencios de los audios "arf_00295_01212219131" y "arf_00295_01212219131_TTS", respectivamente.

con los espectrogramas.

Generalmente, cuando se calcula la STFT y se obtiene la matriz compleja, se aplica una función sobre ella para calcular el módulo y trabajar con valores reales en vez de complejos. Es decir, la STFT produce un array de arrays $X(k) = [X_1(k), X_2(k), ...X_n(k)]$, donde cada uno de esos arrays corresponde al cálculo de la DFT (3.10) realizado sobre una ventana de tamaño m y salto R [17],

$$X_i[k] = \sum_{j=0}^{N-1} f[j]g(j - iR)e^{-2\pi jk/N}$$

que de cara al programa es un array de valores $\omega = a + ib \in \mathbb{C}$. Entonces, se calcula $\sqrt{a^2 + b^2}$ sobre cada elemento obteniendo su módulo, y es esa matriz la que se utiliza usualmente para los espectrogramas.

Sin embargo, como en este trabajo se necesita la parte compleja, a través de la librería *numPy* se pueden separar y mantener ambas partes real y compleja como matrices diferentes, de forma que para la matriz de valores reales se tendría la compuesta por la parte real de los valores ω y, análogamente, los complejos para la matriz de la parte imaginaria. Cada una de estas matrices se puede representar en forma de espectrograma al igual que se hace normalmente con que es es el módulo de los valores complejos.

En los siguientes espectrogramas (figuras 3.5 y 3.6) se puede observar esta

escisión realizada sobre dos señales, de tal forma que aunque los espectrogramas de cada parte son bastante similares al provenir de la misma voz, se pueden observar visualmente diferencias. Esta diferencia de información es la buscamos comprobar si aporta valor a la hora de identificar los *deepfakes*.

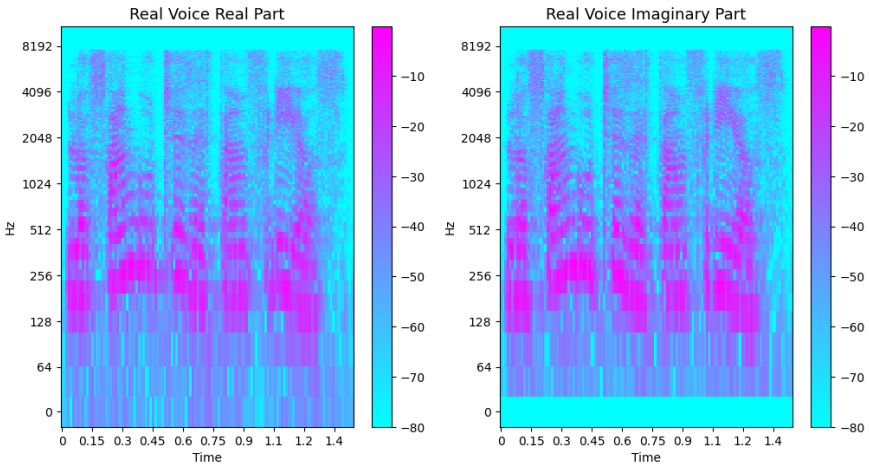


Figura 3.5: Espectrogramas de la parte real y la parte compleja de la señal "arf_00295_01212219131".

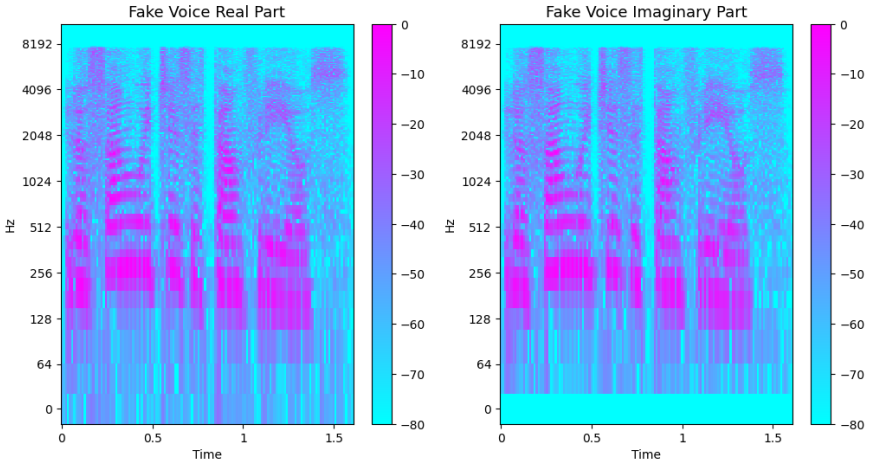


Figura 3.6: Espectrogramas de la parte real y la parte compleja de la señal "arf_00295_01212219131_TTS".

3.2.4. Filtros MEL y MFCC

La escala MEL (Maximally Even and Least) está pensada para tener en consideración cómo escucha el oído humano las voces y melodías y en base a ella se plantearon los filtros MEL. Estos filtros son usados para calcular los Coeficientes Cepstrales en la Escala de Mel o MFCC, una herramienta para la extracción de características del sonido muy utilizado al trabajar con audios de voz o música [18].

Sin embargo, no se hará uso de ellos en este proyecto. La forma de calcular los MFCC incluye también el cálculo del absoluto de la STFT, se obtiene el logaritmo de la señal y finalmente se aplica la Transformada de Coseno Discreta (DCT)[19]. La decisión tomada se fundamenta en que por un lado, se necesitaba que no se calcule el módulo de la señal y se permita seleccionar únicamente la parte compleja, cosa que no se facilita en los métodos usuales para el cálculo de estos coeficientes en los programas informáticos. Por otro lado, como se ha indicado, se realiza el logaritmo de la señal en el proceso, lo que la comprime e impide una visualización correcta de las altas frecuencias. Para el estudio de las características que se verán en la siguiente sección se utilizan una serie de estadísticos, de forma que se consideró que tal vez una correcta visualización de estas frecuencias altas al utilizar la frecuencia lineal en vez de la logarítmica podría proporcionar información que de normal se pasa por alto. Considerando que los filtros MEL simulan la forma de oír de un humano y que para los humanos las voces generadas son casi indistinguibles de las reales, se plantea como un buen entorno de estudio el uso de los espectrogramas usuales sin tener en cuenta la escala MEL.

3.2.5. Características

Para la elección de las características se ha considerado que lo ideal para comenzar un estudio en este área es analizar cómo se comporta la parte compleja con los estadísticos usuales, y estos son, la media, la desviación estándar y una serie de percentiles.
Si bien es cierto que el uso de estos estadísticos depende de la distribución de los datos y por la naturaleza de los los que se tratan en este trabajo parece evidente que no seguirán una distribución normal o siquiera conocida.

La media y la desviación estándar tienen una mayor relevancia si los datos siguen una distribución normal, por lo que se realizaron la prueba de normalización de Kolmogórov-Smirnov, un histograma y un gráfico Q-Q sobre una muestra de los datos para comprobar si siguen este tipo de distribución.

Estas pruebas son de las más usuales para comprobar si unos datos siguen una distribución normal.[20] [21] En cuanto a las pruebas de contraste de hipótesis

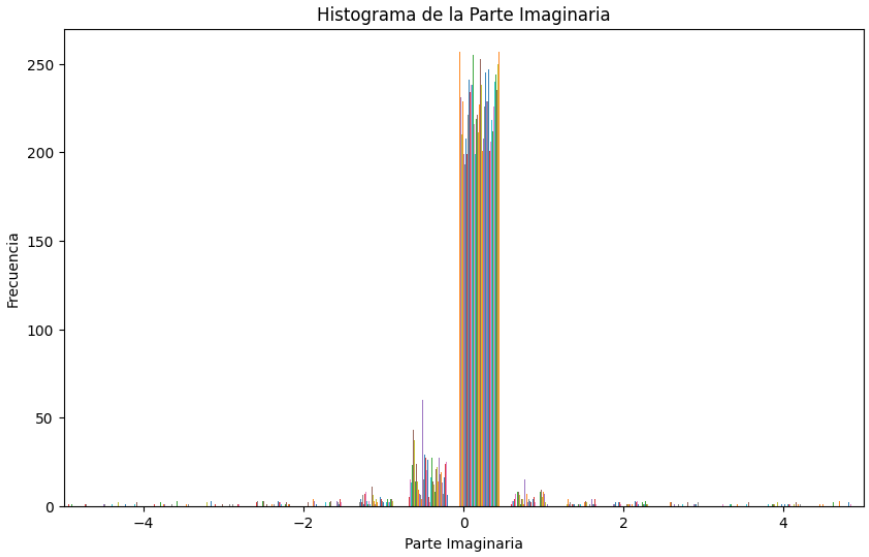


Figura 3.7: Histograma de la parte imaginaria de de la señal "arf_00295_01212219131".

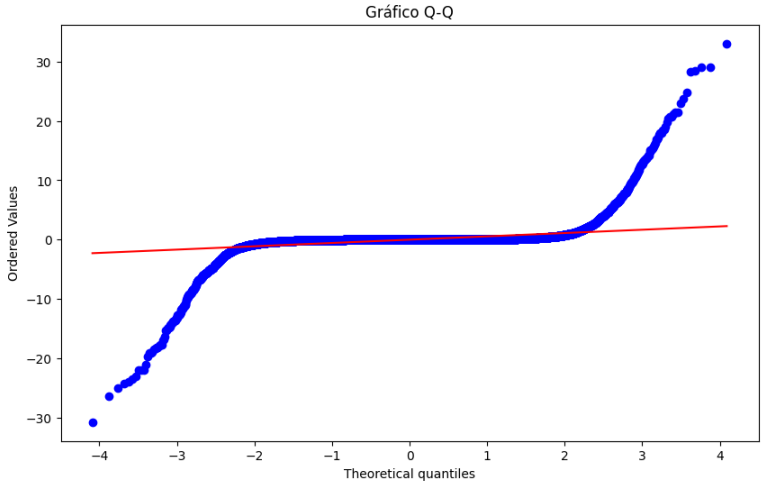


Figura 3.8: Gráfico Q-Q de la parte imaginaria de de la señal "arf_00295_01212219131".

```
Prueba de normalidad de Kolmogorov-Smirnov
Estadístico de KS: 0.3655506374389491
Valor p: 0.0
Se rechaza la hipótesis nula: los datos no siguen una distribución normal.
```

Figura 3.9: Resultado de las prueba de Kolmogórov-Smirnov sobre la parte compleja de la señal "arf_00295_01212219131".

existen varias para determinar si un conjunto de datos sigue una distribución normal; una de las más comunes es la prueba de Shapiro-Wilk [22], pero tiene el inconveniente de que se deja de recomendar una vez el conjunto de datos supera los 50 elementos. En este caso el conjunto tiene $257x125 = 32{,}125$ elementos, por lo que se aleja bastante del dominio de aplicación de esa prueba. Por eso mismo se realizó la prueba de Kolmogórov-Smirnov[23] que sí se utiliza para conjuntos más grandes y permite comparar dos distribuciones, por lo que se ha comparado la del conjunto de la señal con la distribución normal. El resultado de la prueba visible en la figura 3.9 nos indica que los puntos del espectrograma no siguen una distribución normal.

Aunque, dado que el conjunto de datos es bastante grande, será suficiente una pequeña desviación de la normal para que la prueba rechace la hipótesis nula e indique que no sigue dicha distribución, por ello también se han realizado las comprobaciones mediante el histograma y el gráfico Q-Q.
En el histograma de la figura 3.7 se ve claramente que los datos no siguen una distribución normal. La distribución parece simétrica, pero no se adapta a la curva normal.

Por otro lado, el diagrama Q-Q de la figura 3.8 representa los cuantiles de la distribución de la muestra frente a los cuantiles teóricos que tendría si siguiese una distribución normal. Se puede observar que casi la mitad de los datos se alejan de la diagonal, por lo que se concluye que efectivamente, tal y como parecía intuitivo, los puntos de los espectrogramas no siguen una distribución normal.

Evidentemente no se revisado la totalidad del dataset y se ha considerado únicamente una pequeña muestra aleatoria, pero la finalidad de este análisis era conseguir un poco de sensibilidad respecto a las distribuciones de los datos, por lo que viendo que en la muestra salen resultados similares y parece lógico pensar que se cumple al menos para una gran mayoría, puede concluir que a priori la media y la desviación estándar no tienen por qué afectar de especial forma a las características que se calculen.

De esta forma, las características que se van a estudiar, siendo N el número total de ventanas, son:

- Media
$$\mu(k) = \frac{1}{N}\sum^{N} A(k)$$
- Desviación estándar
$$\sigma(k) = \sqrt{\frac{1}{N}\sum^{N}(A(k) - \mu(k))^2}$$
- Percentiles 5, 10, 50, 90, y 95

	filename	ImagPartPercentile_5	ImagPartPercentile_10	ImagPartPercentile_50	ImagPartPercentile_90	ImagPartPercentile_95	ImagPartMean	ImagPartStd
	DATA/asvspoof2019_LA/train_dev/arf_00295_01212...	-0.501703	-0.06	0.0	0.055859	0.529107	-0.0	0.678047
	DATA/asvspoof2019_LA/train_dev/arf_00295_01212...	-1.176957	-0.160341	0.0	0.158928	1.075645	0.0	0.638837
	DATA/asvspoof2019_LA/train_dev/arf_00295_00020...	-1.582135	-0.129684	0.0	0.120395	1.857087	0.0	0.697954
	DATA/asvspoof2019_LA/train_dev/arf_00295_00389...	-0.495066	-0.059371	-0.0	0.05981	0.649975	0.0	0.534486
	DATA/asvspoof2019_LA/train_dev/vem_08784_01339...	-0.189194	-0.004314	-0.0	0.003635	0.226406	0.000001	0.333913
	DATA/asvspoof2019_LA/train_dev/TTS-StarGAN-com...	-0.864793	-0.166892	0.0	0.162585	0.715968	-0.000001	0.855949
	DATA/asvspoof2019_LA/train_dev/vem_09697_02056...	-1.250158	-0.153491	0.0	0.144107	1.25004	-0.0	0.665372

Figura 3.10: Cálculo de las características sobre una muestra del Dataset.

Sea $B(k)$ el array ordenado de $A_i(k)$ $i \in \{1...N\}$.

$$Percentil_J(k) = B(k)(\frac{J}{100} \cdot N) \quad J \in \{5, 10, 50, 90, 95\}$$

Para el cálculo de estos estadísticos se parte de la matriz $A(k)$ que resulta de seleccionar la parte compleja de la generada por la STFT. Se ha de obtener un único valor que represente la característica para ese audio, pero no tiene sentido calcular la métrica a la vez sobre toda la matriz y se tiene que decidir si antes se calcula sobre las filas o sobre las columnas. Dado la naturaleza de la matriz que proviene de un elemento físico como es el espectrograma de un audio, se ha considerado realizar el cálculo primero para cada una de las ventanas, o lo que es lo mismo, por columnas, para finalmente calcular el global. Se ha utilizado la librería numPy en todos los casos. [24] [25] [26]

El siguiente paso es calcular estas características para el dataset de entrenamiento, que en este caso son 15.800 ficheros de audio que mantienen la relación "bonafide"-"spoof" original.

Como el estudio de estas características por sí solas sería bastante poco concluyente, se decide que se incluyan a otras características que se suelen utilizar normalmente. Para esto se hace uso de openSMILE (open-source Speech and Music Interpretation by Large-space Extraction) una herramienta para el análisis, procesamiento y clasificación de audio, especialmente orientado a aplicaciones de voz y música, como es el caso de este proyecto.[27]

OpenSMILE

OpenSMILE pone a disposición varios sets de características estandarizados: ComParE_2016, GeMAPS and eGeMAPS, dentro de ellos además hay distintos niveles y versiones y calculan una cantidad diferente de características.
Debido a lo costoso que es computacionalmente la ejecución de OpenSMILE se eligen las características de GeMAPSv01b[28] que son 32 parámetros, agrupables en cuatro conjuntos:

- Parámetros relacionados con la frecuencia

- Parámetros relacionados con la energía y la amplitud
- Parámetros espectrales
- Características temporales

Es un conjunto que abarca bastantes atributos importantes de la señal de audio, como se demostró en publicaciones previas.[28]

De esta forma, se presentas las características de la parte imaginaria que se extrajeron de la STFT, las mismas características pero calculadas sobre la parte real de la transformada y finalmente las obtenidas gracias a OpenSMILE. Combinando estos conjuntos de distintas formas para el entrenamiento del modelo se realizarán los experimentos de la siguiente sección con el fin de ver en qué posición de relevancia quedan las características generadas en este trabajo.

4

Resultados

Dentro de este capítulo se consideran cuatro secciones correspondientes a distintas ejecuciones del experimento, observando los resultados obtenidos en cada una de ellas para así poder realizar los contrastes adecuados.

La primera ejecución tomará el papel de resultados de control y consistirá en simplemente observar el rendimiento del modelo con las características de GeMAPS, tal y como sería durante una ejecución usual del mismo.
El segundo bloque consistirá en el experimento ya sí contemplando las nuevas características de la parte imaginaria.
Se añade un tercer bloque donde también se incluyen las mismas características pero calculadas sobre la parte real de la señal, de forma que se pueda comparar su relevancia con las de la parte compleja.
Un cuarto bloque surge por la necesidad tener resultados y conclusiones más claras y es el experimento con el dataset de características de GeMAPS unidas a las calculadas en el experimento anterior sobre la parte real, descartando las características de la parte compleja.

Una parte esencial para que tenga sentido el proyecto y la comparativa es que las ejecuciones partan de unas condiciones análogas. Con el fin de conseguir ese objetivo, se considera el mismo conjunto de 15.800 audios del que posteriormente se selecciona el subconjunto de entrenamiento, en este caso, consistente en un 80 % del conjunto original, 12.640 pistas de audio, que se seleccionan aleatoriamente pero siempre manteniendo la distribución existente de "spoof" frente a "bonafide". En consecuencia, el conjunto de validación o test consistirá en el 20 % restante, 3.160 audios.

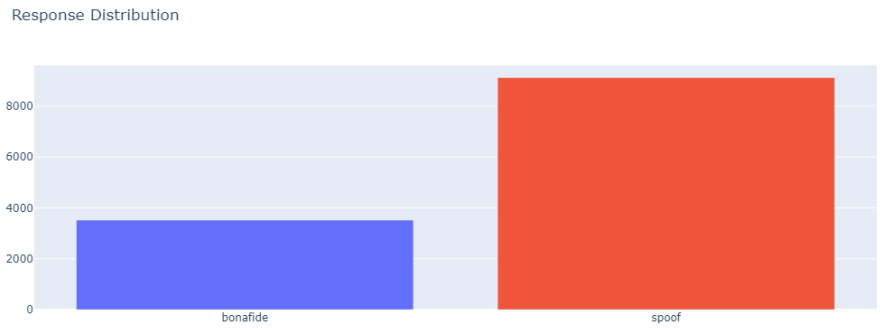


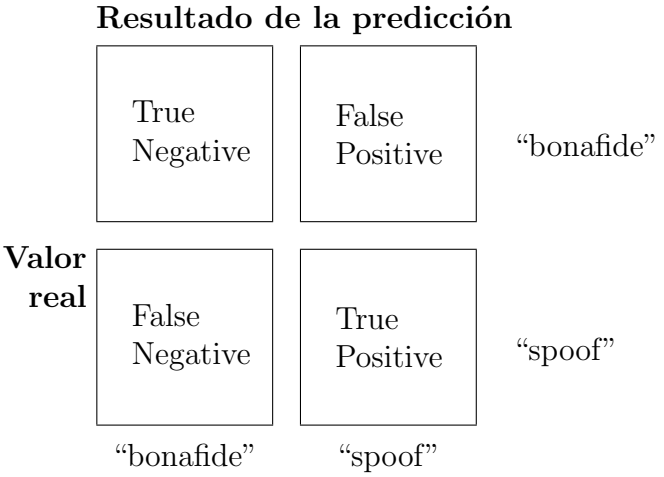
Figura 4.1: Distribución de las muestras dentro del conjunto de entrenamiento.

4.1. Diagramas para el estudio del desempeño del modelo

Cuando se realicen los entrenamientos del modelo en cada una de las ejecuciones, se extraerán una serie de diagramas que permiten comprender el desempeño de cada uno de los modelos. Estos diagramas los genera InterpretML[3], una librería que se encarga de proporcionar "explicaciones" sobre modelos de Machine Learning que permitan al usuario interpretar y comprender en más detalle su funcionamiento. En este trabajo es bastante útil ya que va a permitir observar qué características prioriza el modelo o aquellas que le han sido más útiles, facilitando la extracción de conclusiones. Y junto a él se decide el uso de un Explainable Boosting Machine (EMB)[6], un modelo interpretable con muy buen rendimiento, asemejándose a modelos complejos no interpretables.

4.1.1. Matriz de confusión

Es una representación visual de la clasificación de las voces que ha realizado el modelo, permitiendo un conteo de los aciertos y errores que identifica en cada una de las clases de los datos. En este problema, al tratarse de una cuestión binaria, la matriz de confusión tendrá dimensión (2, 2) pues un audio puede únicamente ser real o falso, y únicamente puede clasificarse en dichas clases.[29] La matriz de confusión de la que se hace uso tiene la siguiente estructura:



Es bastante frecuente ver este tipo de estructuras binarias para las matrices de confusión, pero siempre va a depender del tipo de problema y de la cantidad de clases en las que se pueda clasificar o identificar una muestra, por lo que se puede contar con matrices de dimensión (3,3) o superior.[30]

Las variables que aparecen en el interior de la tabla tabla se explican a continuación:

- **True Positive (Verdadero Positivo, TP):** se refiere al número de clases que han sido identificadas de forma exitosa como pertenecientes a la clase de interés, es decir, cantidad de "spoofs" que han sido detectados.
- **True Negative (Verdadero Negativo, TN):** corresponde con la cantidad de elementos de la clase distinta a la de interés que han sido clasificados correctamente como no perteneciente a ella, en este caso, audios de voces reales que han sido determinados como tal.
- **False Positive (Falso Positivo, FP):** representa el número de clases que han sido identificadas de manera errónea como la clase de interés, o lo que es lo mismo, aquellos audios de voz real clasificados como "spoof". También se les denomina errores de tipo I.
- **False Negative (Falso Negativo, FN):** análogamente a la variable anterior, identifica el número de clases clasificadas de manera errónea, pero esta vez como clases distintas a la de interés, por lo que son aquellos audios "spoof" que han sido identificados como voces reales. También se les denomina errores de tipo II.

Gracias a la agregación de los resultados, se podrán calcular métricas para observar su rendimiento con un mayor detalle y de forma más esclarecedora.

- Accuracy: es la proporción de muestras clasificadas correctamente respecto al total. Indica en términos generales cuánto se aproxima el modelo con sus predicciones.

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

- Precisión (valor predictivo positivo, VPP): es la proporción de elementos verdaderos positivos respecto al número total de predicciones positivas.

$$Precisión = \frac{TP}{TP + FP}$$

- Sensibilidad (recall o tasa de verdaderos positivos, TVP): es la proporción entre el número de casos positivos que fueron identificados correctamente con respecto al número total de casos positivos.

$$Sensibilidad = \frac{TP}{TP + FN}$$

- Especificidad (tasa de verdaderos negativos, TVN): es la proporción entre el número de casos negativos que fueron identificados correctamente y el número total de casos negativos.

$$Especifidad = \frac{TN}{TN + FP}$$

- F1 Score: se trata de la media armónica de la precisión y la sensibilidad, combinando ambas métricas en una única. Es de gran utilidad cuando el conjunto de datos no está balanceado.

$$F1 \quad Score = \frac{2 \cdot VPP \cdot TVP}{VPP + TVP}$$

4.1.2. Diagrama de la relevancia de las características

InterpretML proporciona un diagrama de barras donde muestra las 15 características o combinaciones de estas que más relevancia han tenido a nivel global para clasificar las voces en reales o falsas.

Puede ser que una característica por sí sola proporcione bastante información, por ejemplo, si a partir de cierto valor predomina con diferencia en las voces reales, le va a servir como gran determinante al modelo para hacer la clasificación. Sin embargo, no sólo se tienen en cuenta las características de forma aislada, ya que considerando relaciones entre ellas es más frecuente encontrar patrones que permitan hacer una clasificación. Si sólo fuera necesario tener en cuenta las características por sí solas, entonces es probable que, viendo la lista

de valores de las mismas para un fichero y sabiendo unas condiciones, incluso un humano pudiera hacer la clasificación. Evidentemente no es tan sencillo y por eso se hace uso de los modelos de aprendizaje automático para alcanzar el objetivo. El modelo se encarga de extraer estas relaciones y combinaciones que ha encontrado importantes, pudiendo observarlas en el ranking que nos proporciona InterpretML.

Además de este tipo de gráfica global, también se incluyen unas individuales para cada característica y que muestra la relación entre los valores de la característica y su impacto en la predicción del modelo. En estos diagramas se observa un gráfico de lineas que, en el caso concreto de los modelos clasificatorios, se representa para cada valor de la característica la probabilidad logarítmica de que determine la voz como "spoof" frente a que no; de forma que es interesante ver el tipo de curva que se dibuja. Si resulta en una línea plana, quiere decir que esa característica no aporta información relevante, mientras que si se encuentra una pendiente no nula, ya sea ascendente, descendente o variable, es más probable que esa característica haya sido relevante para el modelo. [31]

Se ha de indicar que se han realizado varios reentrenamientos con conjuntos de entrenamiento y test diferentes en un mismo experimento para observar si de alguna forma el ranking podía verse afectado, pero siempre han mantenido el mismo orden, por lo que supone un medio para estudiar las características y su interés de forma sólida.

4.2. Experimento base

Al observar la matriz de confusión de la figura 4.2, puede observarse que se consiguen las siguientes métricas:

Métrica	**Accuracy**	**Precisión**	**Sensibilidad**	**Especifidad**	**F1 Score**
Valor	0.9158	0.9357	0.9488	0.8295	0.9226

Tabla 4.1: Métricas de rendimiento del experimento base.

Se puede ver que el modelo no es tan específico como sensible y que es más probable que falle determinando un Falso Positivo, pero en cualquier caso, son resultados bastante robustos. [32]

Por otro lado, cuando se analiza la figura 4.3, se puede observar que las características que tienen una mayor relevancia son:

1. "slopeV0-500_sma3nz_amean" es el valor de la media suavizada de cómo varía la señal a lo largo del tiempo entre 0Hz y 500Hz en base a la pendien-

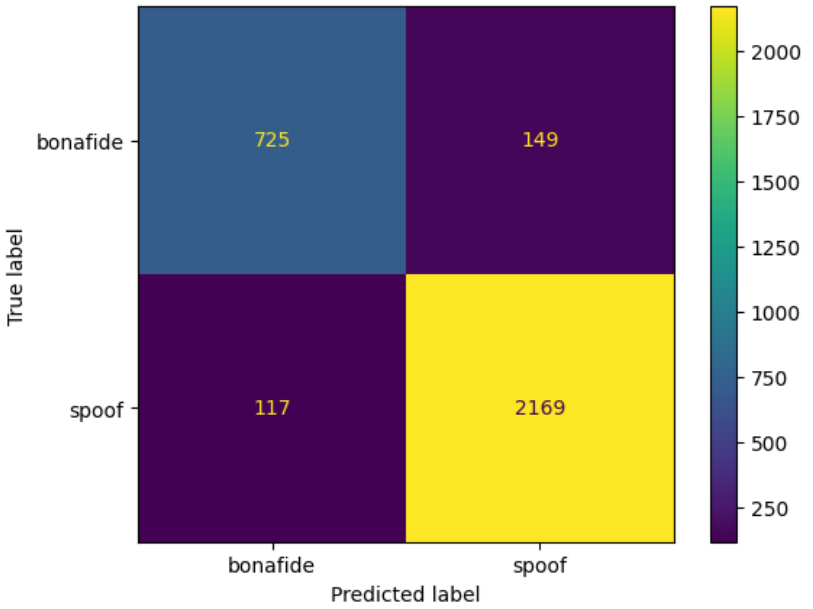


Figura 4.2: Matriz de confusión del experimento base.

te (slope). Pertenece al grupo de parámetros relacionado con el balanceo espectral.

2. "slopeUV0-500_sma3nz_amean" es análoga a la característica anterior pero en segmentos donde no hay voz, y por lo tanto también pertenece al mismo grupo de parámetros.

3. "loudness_sma3_percentile20.0" indica el percentil 20 de la sonoridad, o volumen, de la muestra y está dentro del grupo de parámetros relacionados con la energía.

4. "shimmerLocaldB_sma3nz_amean" es la media suavizada de la inestabilidad o variación de la amplitud de la señal en el tiempo. Se encuentra en el grupo de parámetros relacionados con la amplitud.

5. "logRelF0-H1-A3_sma3nz_amean" representa la media de la relación logarítmica entre la frecuencia fundamental (F0) y las amplitudes de los primeros armónicos (H1, primer armónico, y A3, tercer armónico) y que generalmente se usa para medir la claridad de la voz. Pertenece al grupo de parámetros relacionados con la frecuencia.

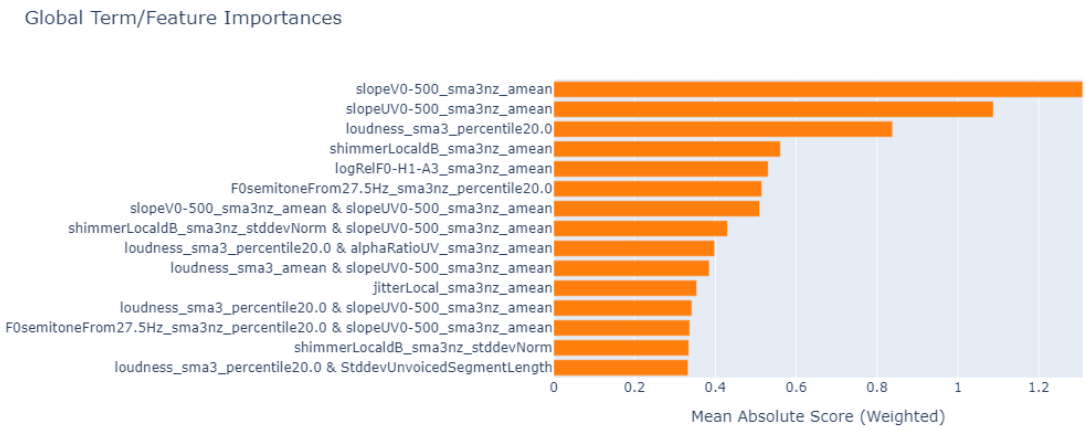


Figura 4.3: Ranking de características del experimento base.

4.3. Experimento con las características de la parte imaginaria

Recuérdese que este experimento se ha realizado de la misma forma que el anterior, con la misma configuración a la hora de generar los conjuntos de entrenamiento y validación y la única diferencia de que el dataset que se le pasa ya no incluye sólo las características de GeMAPS sino también siete columnas nuevas "ImagPartPercentile_5", "ImagPartPercentile_10", "ImagPartPercentile_50", "ImagPartPercentile_90", "ImagPartPercentile_95", "ImagPartMean", "ImagPartStd".
Observando la matriz de confusión de la figura 4.4, se calculan las siguientes métricas:

Métrica	**Accuracy**	**Precisión**	**Sensibilidad**	**Especifidad**	**F1 Score**
Valor	0.9484	0.9557	0.9737	0.8821	0.9646

Tabla 4.2: Métricas de rendimiento del primer experimento.

Todas las métricas han mejorado en torno al 3 %. Al analizar la figura 4.5, es notable que esta mejora se debe a que algunas de las métricas nuevas han sido útiles para el modelo. Las principales en este caso han sido:

1. "ImagPartStd" es la desviación típica que se ha calculado sobre la parte imaginaria de la señal.

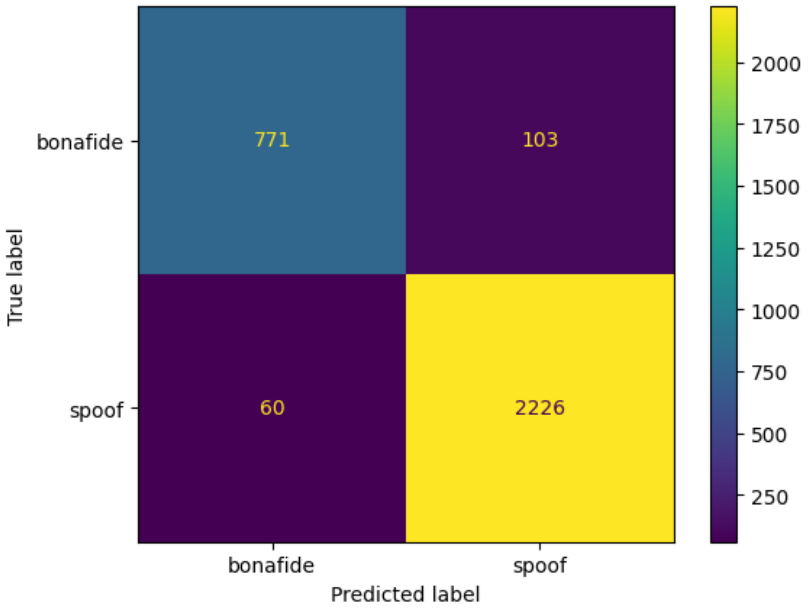


Figura 4.4: Matriz de confusión del primer experimento.

2. "slopeV0-500_sma3nz_amean", era la que en el experimento 0 estaba en el primer puesto de relevancia.

3. "ImagPartMean" es la mediana calculada sobre la parte imaginaria de la señal.

4. "slopeUV0-500_sma3nz_amean" ha pasado del segundo puesto de relevancia al cuarto.

5. "loudness_sma3_percentile20.0" ha pasado del tercer puesto de relevancia al quinto.

Destacar también que en el puesto 11. se encuentra la combinación "slopeUV0-500_sma3nz_amean & ImagPartMean"; en el 13., la combinación "ImagPartStd & ImagPartMean" y en el 15., "slopeUV0-500_sma3nz_amean & ImagPartStd". Siendo indicativo de que las características de la media y la desviación típica de la parte compleja de la señal son relevantes para el modelo tanto de forma aislada como en combinaciones de ellas con ellas o con "slopeUV0-500_sma3nz_amean".

Las características referentes a los percentiles aparentemente no aportan información suficiente como para que tengan un gran peso a la hora de clasificar una voz y siquiera aparecen en el diagrama.

A continuación se presentan las gráficas individuales de las características que están en entre las principales, "ImagPartStd" y "ImagPartMean", para entender con un mayor detalle la relevancia que han tenido.

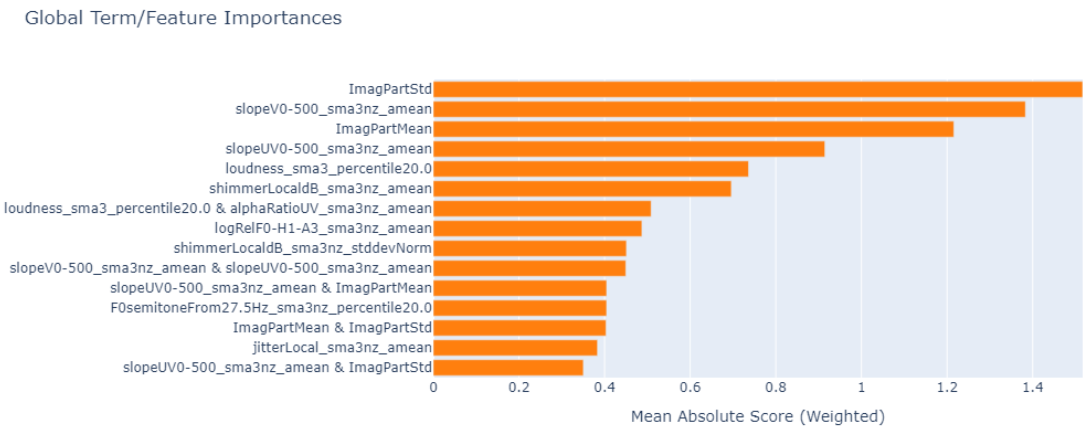


Figura 4.5: Ranking de características del primer experimento.

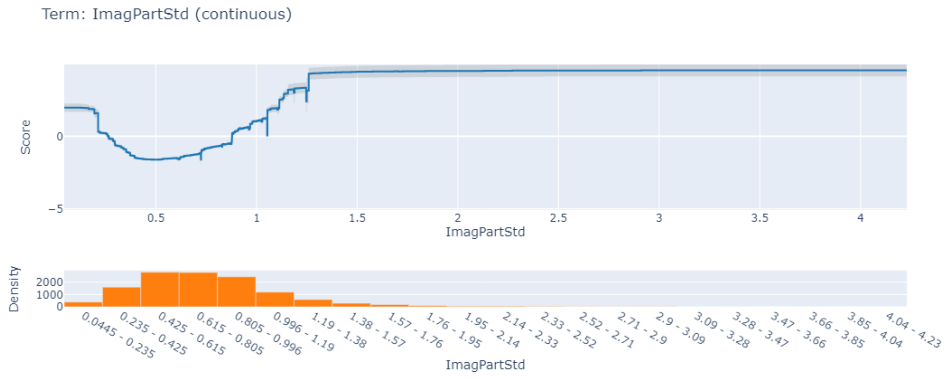


Figura 4.6: Diagrama de desempeño de la característica "ImagPartStd" en el primer experimento.

En el Apéndice A.1 se podrán encontrar el resto de gráficas individuales para las otras características calculadas sobre la parte imaginaria, y donde se podrá comprobar que efectivamente tienen un diagrama mucho más plano corroborando el hecho de que el modelo no ha encontrado relación destacable alguna con la predicción para estas características.

En la figura 4.6 correspondiente a la característica "ImagPartStd", se perciben dos pendientes: una decreciente a medida que el valor de la desviación típica crece hasta 0.5 y otra creciente junto con el valor del atributo a partir de 0.5, lo cual indica que es más probable que la muestra sea positiva (spoof) mientras más se aleja de ese valor.

Obsérvese en el histograma 4.7 que los valores de "bonafide" en azul se distribuyen principalmente donde la curva del diagrama de desempeño decae, mien-

tras que para los casos falsos, en naranja, se observa que gran parte se encuentra acercándose al valor 1 o sobrepasándolo. Esta distribución en los valores hace evidente la decisión del modelo y por qué considera que es tan probable que un fichero sea falso una vez sobrepasa aproximadamente el 0,8. Para clarificar, también se puede observar el diagrama de densidad correspondiente en la figura 4.8.

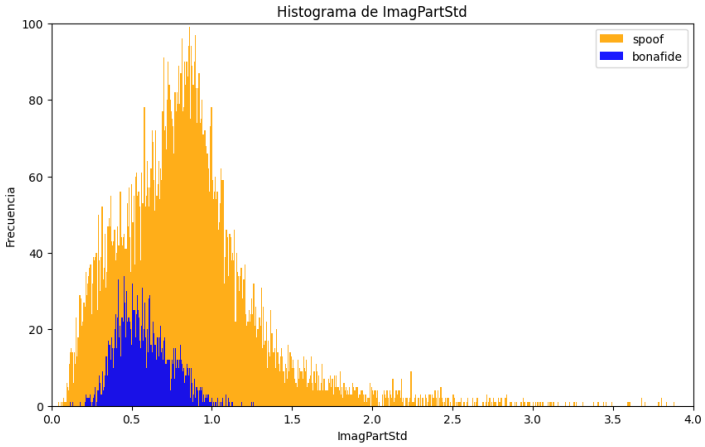


Figura 4.7: Histograma de la distribución de la característica "ImagPartStd".

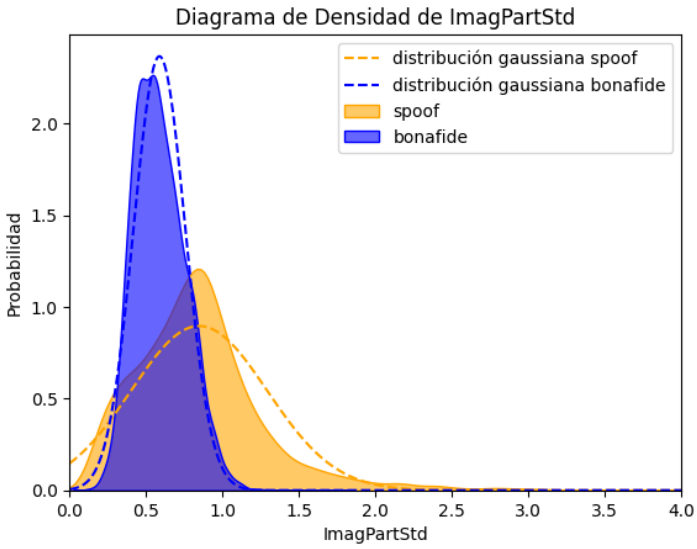


Figura 4.8: Diagrama de densidad de la distribución de la característica "ImagPartStd".

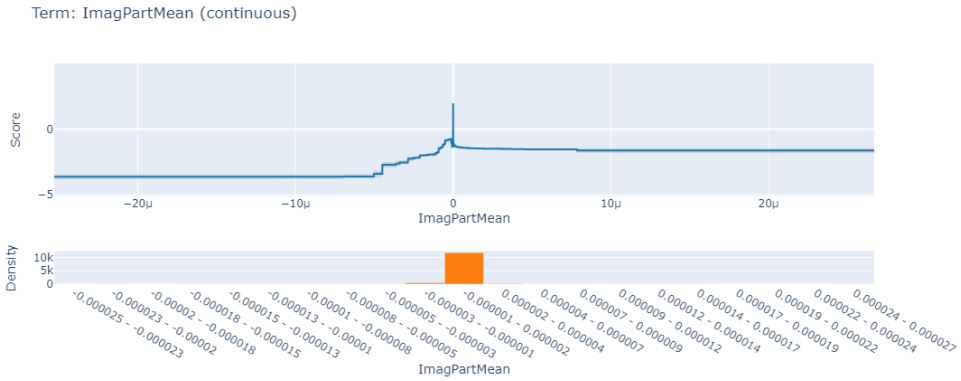


Figura 4.9: Diagrama de desempeño de la característica “ImagPartMean” en el primer experimento.

En la figura 4.9 correspondiente a la característica “ImagPartMean” se destaca que tiene menos dispersión y por ello en gran parte del diagrama se presenta la curva con pendiente nula, pero en el valor nulo la probabilidad de que detecte esa voz como falsa es bastante alta y se puede ver un pico. Con la pendiente creciente en los valores negativos cercanos a cero, parece indicar que las voces reales presentan medias más alejadas del cero y a medida que esta métrica se acerca a ese valor es más probable que la muestra sea “spoof”.

Al observar en este caso el histograma correspondiente presente en la figura 4.10, sacar una relación con el diagrama de rendimiento no es quizá tan evidente, pero comparando las curvas de uno y otro es bastante más probable que en el cero o cercano a él en la parte negativa sea “spoof” ya que en ese valor casi que se concentran todos los elementos, llegándose a asemejar a una distribución de Delta de Dirac[33]. Este distribución también se encontrará presente en la referente a “RealPartMean” como se puede ver en la figura 4.15.
Por el otro lado, la parte de “bonafide” tiene una mayor dispersión y por tanto tienen más probabilidad de darse esta categoría en valores no nulos. Esta dispersión, aunque se puede percibir en el histograma, se contempla mejor en el diagrama de densidad de la figura 4.11.

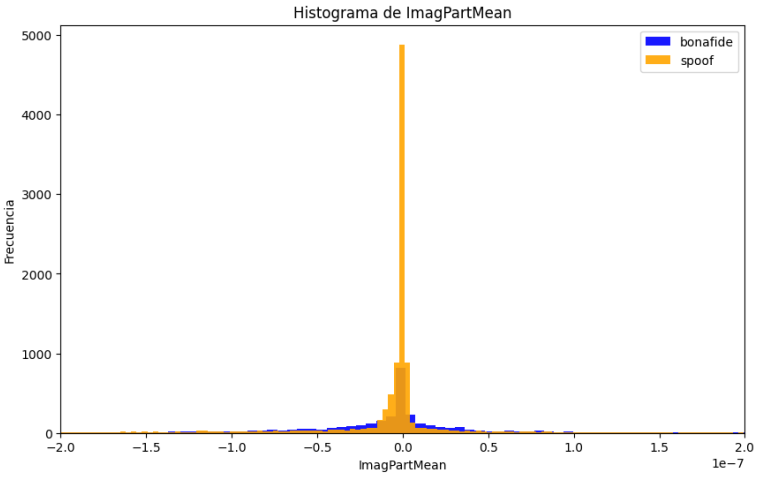


Figura 4.10: Histograma de la distribución de la característica "ImagPartMean".

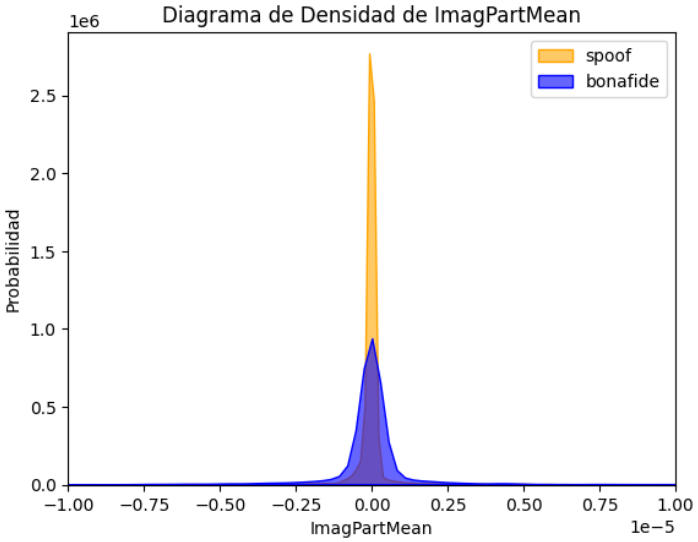


Figura 4.11: Diagrama de densidad de la distribución de la característica "ImagPartMean".

4.4. Experimento con las características de la parte imaginaria y la real

Análogamente al experimento 1, en este lo único que varía es que el dataframe que contiene las características que va a utilizar el modelo contiene las características obtenidas de GeMAPS, las generadas sobre la parte compleja y los mismos estadísticos calculados sobre la parte imaginaria pero generados sobre la parte real de la señal de audio.

Nuevamente, gracias a la matriz de confusión de la figura 4.12 se obtienen las métricas:

Métrica	**Accuracy**	**Precisión**	**Sensibilidad**	**Especifidad**	**F1 Score**
Valor	0.9810	0.9839	0.9899	0.9576	0.9868

Tabla 4.3: Métricas de rendimiento del segundo experimento.

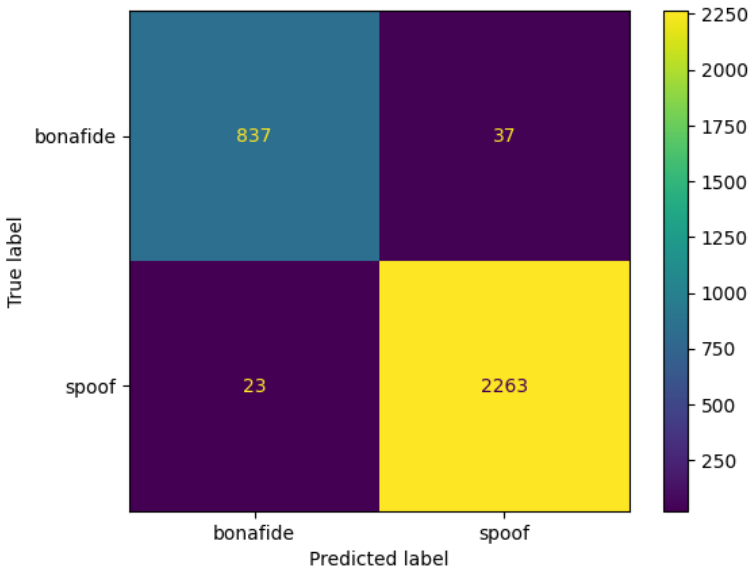


Figura 4.12: Matriz de confusión del segundo experimento.

Y sobre el diagrama de relevancias de la figura 4.13 las características de los primeros puestos son:

1. "RealPartMean" es la media que se ha calculado sobre la parte real de la señal.

2. "ImagPartMean", ha pasado de tercer puesto de relevancia a segundo.

3. “slopeUV0-500_sma3nz_amean” ha subido de cuarto puesto en el orden a tercero..

4. “ImagPartStd” ha pasado de primer orden de relevancia al cuarto.

5. “slopeV0-500_sma3nz_amean & RealPartMean” una combinación nueva al introducir la media de la parte real.

A parte de estas características nuevas entre las cinco más relevantes, sólo se encuentran la combinación “slopeUV0-500_sma3nz_amean & ImagPartMean” en el top 13., y en el top 14., “loudness_sma3_percentile20.0 & ImagPartStd”, pero tampoco parecen tener relevancia en este caso los percentiles de ninguna de las partes.

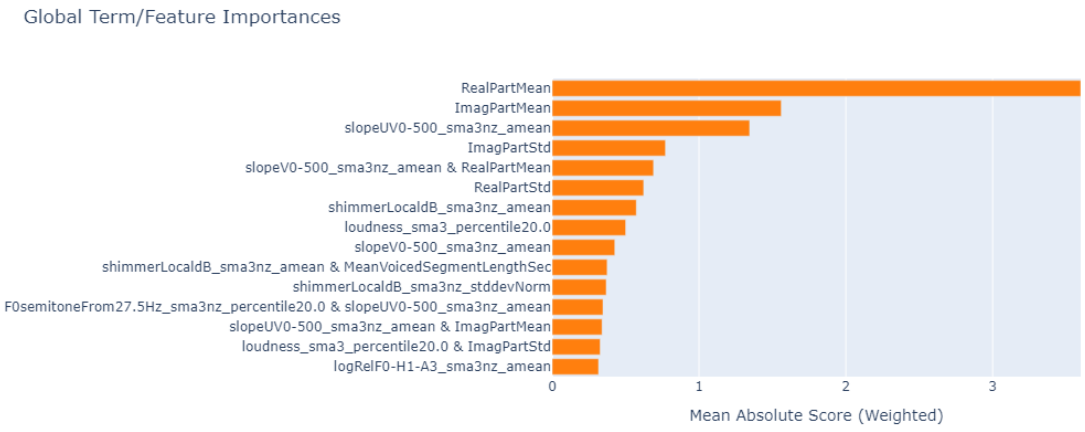


Figura 4.13: Ranking de características del segundo experimento.

Se procede a analizar las gráficas individuales para los atributos “RealPartMean”, “ImagPartMean” y “ImagPartStd”. Las gráficas para el resto de atributos se pueden consultar en el Apéndice A.2

En la figura 4.14 correspondiente a la característica “RealPartMean”, nótese que la probabilidad de que una muestra sea “spoof” cuando la media de la parte real es 0, es prácticamente nula, y que por el contrario alcanza una probabilidad muy alta cuando se aleja de este valor y si se aleja tomando valores positivos.

Si se analiza el histograma correspondiente a los valores de esta característica, como se ve en la figura 4.15, los valores de “bonafide” se agrupan, sin apenas dispersión, en los valores negativos cercanos a cero, justo donde cae en pico la probabilidad de que sea “spoof” en el gráfico 4.14. Para más detalle, debido a que en el histograma comparativo es bastante sutil, obsérvense las figuras 4.16 y 4.17 correspondientes al diagrama de densidad y al histograma de valores “bonafide”,

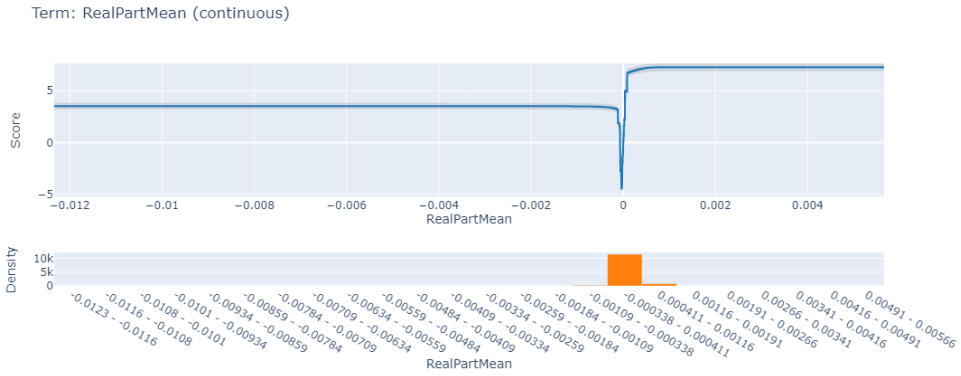


Figura 4.14: Diagrama de desempeño de la característica "RealPartMean" en el segundo experimento.

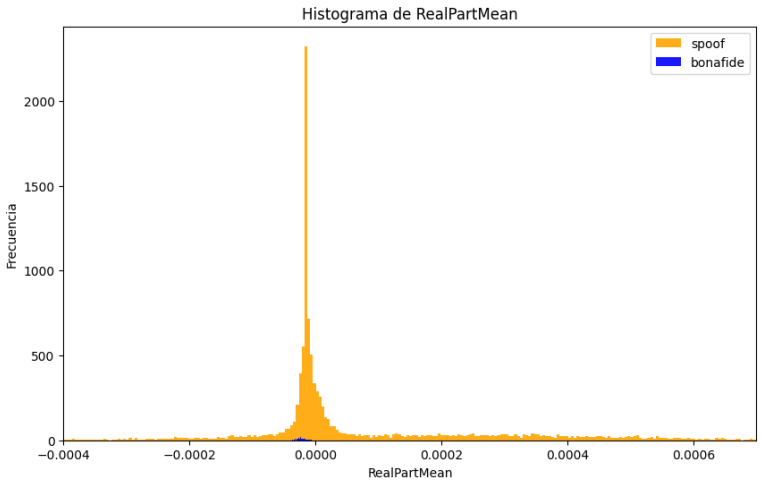


Figura 4.15: Histograma de la distribución de la característica "RealPartMean".

respectivamente. Se puede observar lo que anteriormente se indicaba, siguen una distribución de forma muy concreta y poco dispersa y por ello es lógico que el modelo dé prioridad a clasificar la voz negativamente cuando se encuentra estos valores.

De nuevo, en la figura 4.18 correspondiente a la característica "ImagPartMean", se comprueba que el comportamiento es similar al que tuvo lugar en el experimento 1, con la figura 4.9.

En esta ocasión, en la figura 4.19 correspondiente a la característica "ImagPartStd", aunque al igual que ha ocurrido con "ImagPartMean" y el comportamiento es similar al que tuvo en el experimento 1 (figura 4.6), es destacable la forma en que la curva se ve suavizada y con pendientes menores.

Debido a que este experimento ha tenido mejores resultados que el anterior

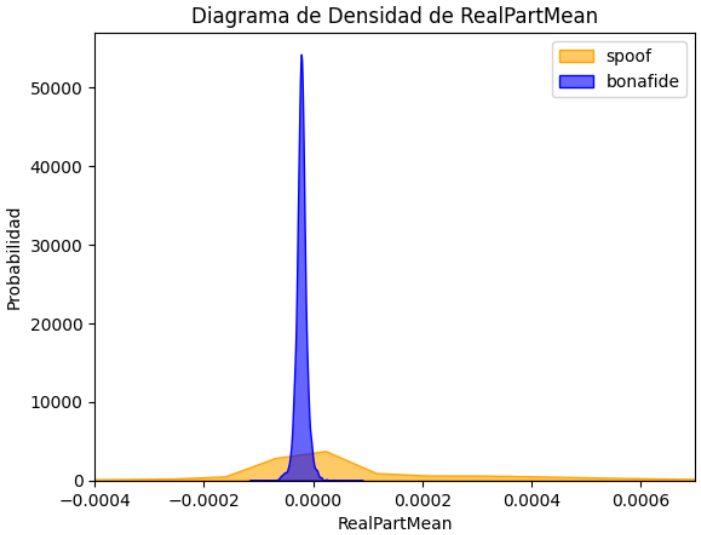


Figura 4.16: Diagrama de densidad de la distribución de la característica "RealPartMean".

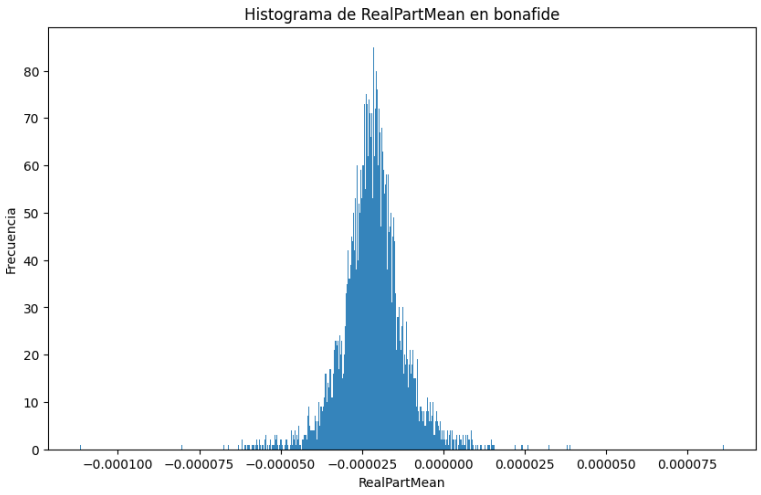


Figura 4.17: Histograma de la distribución de la característica "RealPartMean" para los valores "bonafide".

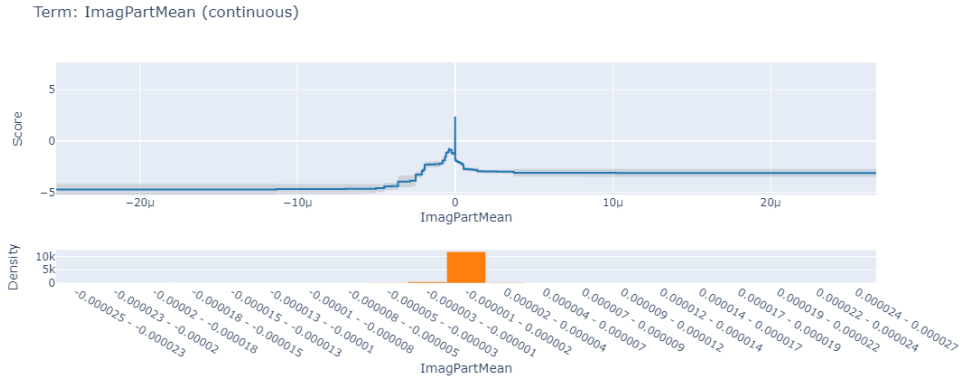


Figura 4.18: Diagrama de desempeño de la característica “ImagPartMean” en el segundo experimento.

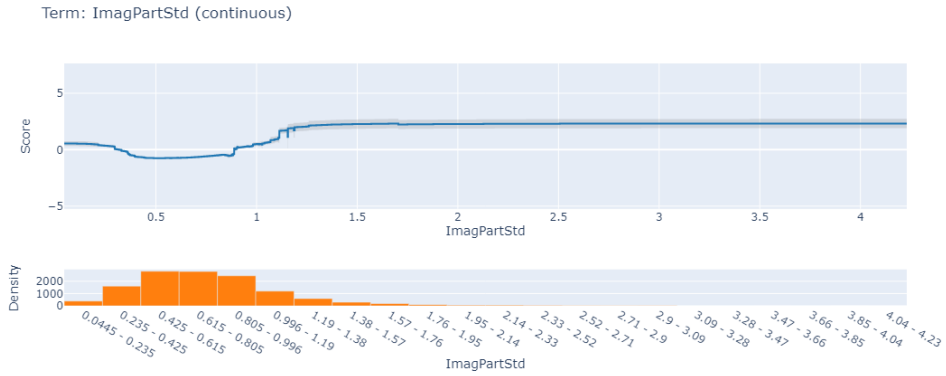


Figura 4.19: Diagrama de desempeño de la característica “ImagPartStd” en el segundo experimento.

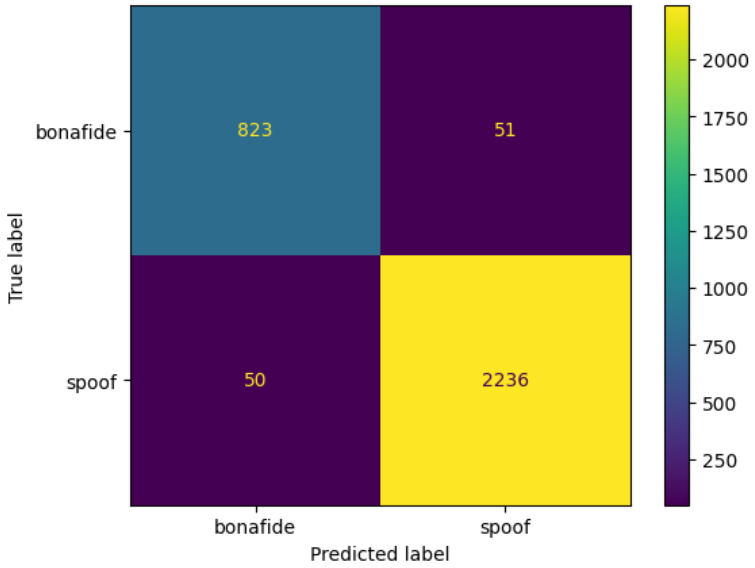


Figura 4.20: Matriz de confusión del tercer experimento.

donde sólo se incluían las características de la parte compleja, se decide hacer un cuarto experimento donde únicamente se incluyan las características de GeMAPS y las calculadas sobre la parte real, para observar el desempeño de ese modelo.

4.5. Experimento con las características de la parte real

En esta ocasión la matriz de confusión de la figura 4.20 proporciona las siguientes métricas:

Métrica	**Accuracy**	**Precisión**	**Sensibilidad**	**Especifidad**	**F1 Score**
Valor	0.9680	0.9777	0.9781	0.9416	0.9778

Tabla 4.4: Métricas de rendimiento del tercer experimento.

Del diagrama de relevancias de la figura 4.21 se extraen las características de los primeros puestos:

1. "RealPartMean" al igual que en el experimento anterior está en el primer puesto.
2. "slopeUV0-500_sma3nz_amean" se encuentra en el segundo lugar al igual que ocurría en el primer experimento.

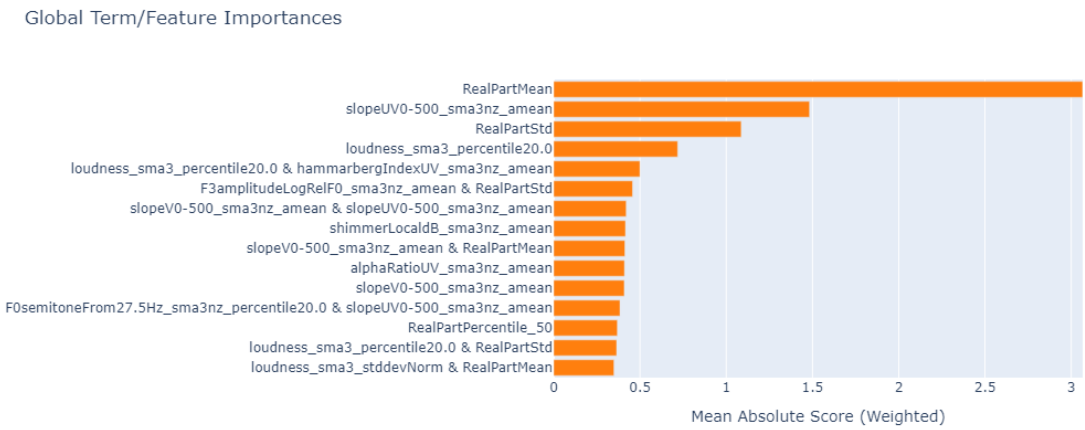


Figura 4.21: Ranking de características del tercer experimento.

3. "RealPartStd", es la desviación típica calculada sobre la parte real.
4. "loudness_sma3_percentile20.0" aparece igual que en experimento base.
5. "loudness_sma3_percentile20.0 & hammarbergIndexUV_sma3nz_amean" una nueva combinación entre la sonoridad y la media del índice de Hammarberg en tramos sin voz.

En este caso sí que se encuentran más características de las calculadas sobre la parte real entre las quince más relevantes, ya sea solas como "RealParPercentile_50" en el puesto 13. o en combinaciones: "F3amplitudeLogRelF0_sma3nz_amean & RealPartStd" (6.), "slopeV0-500_sma3nz_amean & RealPartMean" (9.), "loudness_sma3_percentile20.0 & RealPartStd" (14.) y "loudness_sma3_stddevNorm & RealPartMean" (14.)

Se procede a analizar las gráficas individuales para los atributos "RealPartMean" y "RealPartStd". Las gráficas para el resto de atributos se pueden consultar en el Apéndice A.3

En la figura 4.22 correspondiente a la característica "RealPartMean", se observa un comportamiento prácticamente igual al que tuvo en el segundo experimento, incluso a cuanto nivel de probabilidad.

La figura 4.23 correspondiente a la característica "RealPartStd", tiene un diagrama muy similar al que se encontraba con "ImagPartStd" en el primer experimento (véase la figura 4.6) y no son opuestos como sucede con las gráficas de la media en la parte real e imaginaria. En este caso, se ve una franja grisácea rodeando la línea azul y es lo que se determina "error bar" y es indicativo de que el modelo puede ser inestable en ese área por falta de valores. [34]

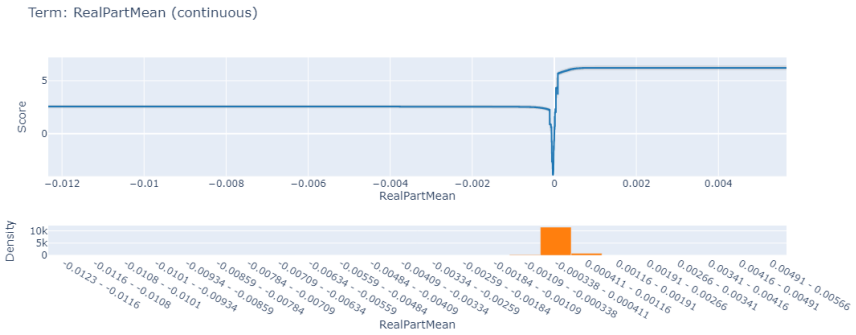


Figura 4.22: Diagrama de desempeño de la característica “RealPartMean” en el tercer experimento.

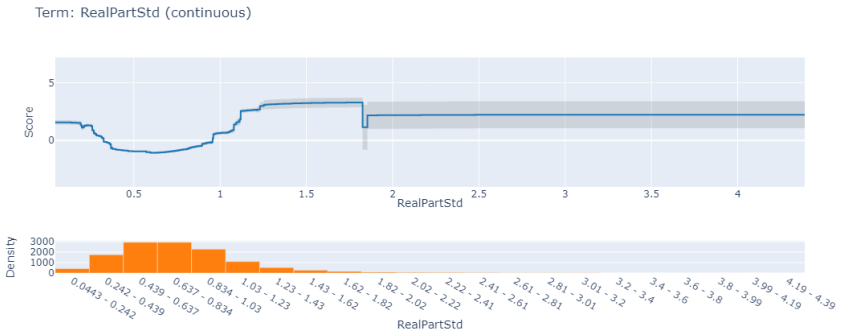


Figura 4.23: Diagrama de desempeño de la característica “RealPartStd” en el tercer experimento.

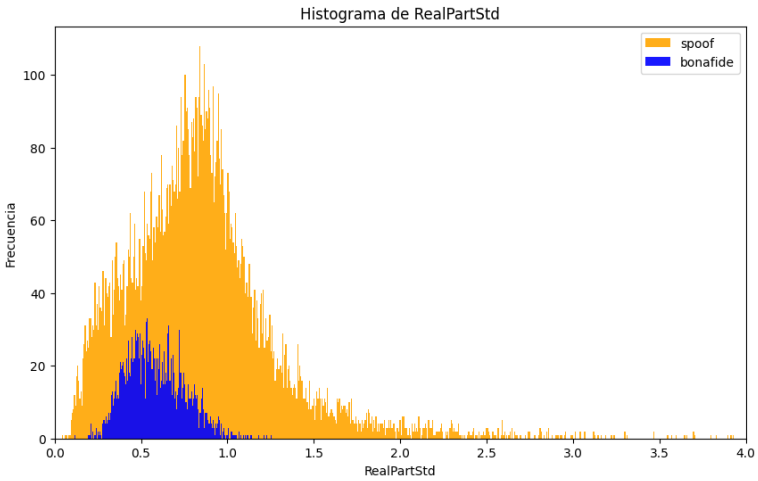


Figura 4.24: Histograma de la distribución de la característica "RealPartStd".

De igual forma, cuando se observa en la figura 4.24 el histograma correspondiente a "RealPartStd" y en la 4.25 el diagrama de densidad, se comprueba que ha sido un comportamiento análogo al de "ImagPartStd" en cuanto a distribución por clases "spoof" y "bonafide" (figuras 4.7, 4.8) y que por lo tanto es presumible que la curva de desempeño proporcionada por el modelo sea también bastante similar.

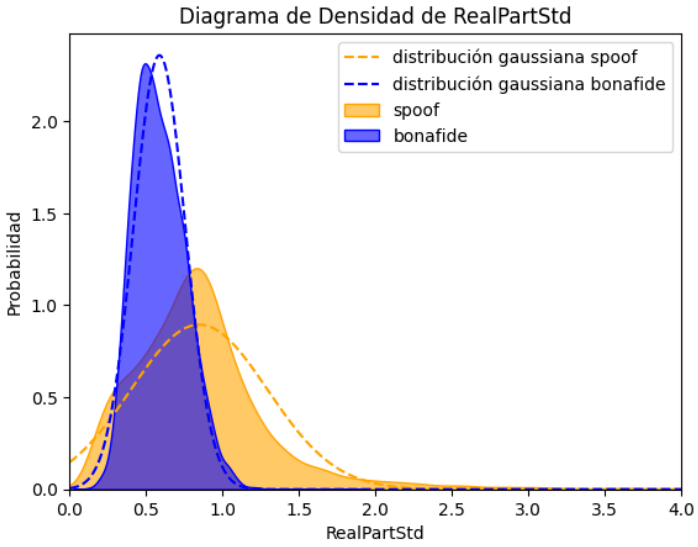


Figura 4.25: Diagrama de densidad de la distribución de la característica "RealPartStd".

4.6. Comparación de resultados

Finalmente, tras haber comprobado en las secciones previas los resultados de los experimentos en detalle, se presenta a continuación una tabla comparativa con las métricas de rendimiento de cada uno de ellos:

Experimento	**Acccuracy**	**VPP**	**TVP**	**TVN**	**F1 Score**
Experimento 0	91.58 %	93.5 %	94.88 %	82.95 %	92.26 %
Experimento 1	94.84 %	95.57 %	97.37 %	88.21 %	96.46 %
Experimento 2	**98.10 %**	**98.39 %**	**98.99 %**	**95.76 %**	**98.68 %**
Experimento 3	96.80 %	97.77 %	97.81 %	94.16 %	97.78 %

Tabla 4.5: Comparación de las métricas de rendimiento entre los diferentes experimentos.

Se puede observar que el modelo ha conseguido un mejor rendimiento globalmente a cada experimento que se ha realizado respecto al experimento base o de control. A pesar de que el experimento con las características de la parte compleja había dado buenos resultados, al unirle también las características de la parte compleja tuvo un mejor rendimiento acercándose en casi todas las métricas al 99 % de éxito.
Con el interés de saber cuánto habrían aportado las características de la parte real se realizó un tercer experimento donde el dataset de características consistía únicamente en las de GeMAPS y las calculadas en la parte real, sin tener en cuenta aquellas de la parte imaginaria. En esta ocasión, no se alcanzaron resultados tan buenos como los obtenidos en el segundo experimento, pero igualmente superaron a los del primer experimento con únicamente la parte compleja de la señal.

Tanto en el primer experimento como en el segundo, la media y la desviación típica de la parte imaginaria han tenido un papel muy relevante en la detección de voces falsas. En el segundo experimento, además, la media de la parte real tomó una posición principal en el modelo, al igual que sucedió en el tercer experimento. Algo que parece interesante a destacar es que la gráfica de ambas medias de la parte real y la parte compleja son, de forma relativamente vaga, simétricas impares respecto al eje X. De tal forma que si en la media de la parte real el valor es nulo supone un indicativo de que la muestra es real, cuando en la parte compleja es nulo, supone lo contrario y lo más probable es que sea "spoof". Y, sin embargo, con las gráficas de la desviación típica, no sucede lo mismo; y es más, al juntarlas en el segundo experimento, parece ser que al existir cierta correlación, se prescinde de una de ellas y prevalece la de la parte imaginaria por tener una curva con mayores pendientes

En cualquier caso, el experimento que tuvo mejor desempeño ha sido el que

contaba con las características de la parte real y de la parte imaginaria, parece lógico deducir que se debe a que une características que aportan mucha información (como se observa en los experimentos 1 y 3) pero no la misma. Así, se considera la parte imaginaria de la señal de voz, y en concreto la separación de la señal en sus componentes reales y complejas, un recurso a tener en cuenta a la hora de clasificar voces falsas por sus buenos resultados en este primer estudio, seguramente debiéndose ellos a que la parte imaginaria y las características aquí calculadas están relacionadas con la fase de la señal de audio que no suele tenerse en cuenta con los algoritmos de generación de voz.

5

Conclusiones

Para la realización de este trabajo era indispensable comprender los fundamentos teóricos que se encuentran tras los medios que se utilizan actualmente para trabajar con las voces. Las librerías que se utilizan para trabajar con los modelos de entrenamiento ya sea supervisado o no, están bastante pulidas para facilitar su uso y es complejo realizar modificaciones de aspectos más esenciales sobre ellas.
Por ejemplo, al igual que la librería *librosa* permite realizar la STFT para hallar el espectrograma normal, también se puede utilizar la librería para calcular los espectrogramas MEL, pero como se indicaba en la sección 3.2.4, debido a que no permite la modificación interna para diferenciar parte real y parte imaginaria, no era conveniente su uso en este trabajo, procediendo esta conclusión del entendimiento de los fundamentos que se hallan tras las herramientas que se utilizan para el procesamiento de la voz.

A lo largo del capítulo 3.1 se estudió la parte teórica necesaria para extraer la parte compleja de la señal y generar los espectrogramas complejos con ella.
Se comenzó en la sección 3.1.1 desarrollando las series temporales y viendo que en concreto las Series de Fourier y su transformada son la base para representar matemáticamente las voces mediante funciones. Sabiendo esto, se empezó estudiar cómo aplicar esta teoría a casos prácticos a lo largo de la sección 3.1.3 de forma concluyeron los algoritmos más usuales para su cálculo como la Transformada Discreta de Fourier (DFT 3.1.3), la Transformada Rápida de Fourier (FFT 3.1.3) y, la que finalmente se utiliza en este proyecto, la Transformada Rápida de Fourier (STFT 3.1.3).

Con esto finalizaba el cumplimiento del objetivo 2.1 referente a la modelización de la parte compleja de la señal de voz gracias a la Transformada de Fourier y sus aplicaciones.

Una vez se disponían de las herramientas para la parte práctica de este trabajo, se comenzó con la primera parte referente a cualquier proceso de entrenamiento de un modelo: selección y tratamiento de los datos a utilizar. En las secciones 3.2.1 y 3.2.2 se elegían y trataban los audios de voz del dataset HABLA por ser una gran base de datos en español con voces tanto reales como generadas mediante distintas técnicas de spoofing, y en la sección 3.2.3 se aplicaban las herramientas conseguidas con la transformada y los espectrogramas para extraer la parte compleja de las señales de voz ya tratadas.

Todo este proceso sirvió de base para el cumplimiento del subobjetivo 2.2-1 consistente en seleccionar qué propiedades se estudiarían de la parte compleja de la señal y que finalmente tuvo su resolución en el apartado 3.2.5 cuando se decidieron estudiar los estadísticos.

Luego, en la sección 3.2.5 se presenta el dataset de características que permitirá la comparación con las que se han calculado "manualmente" en este proyecto. El dataset elegido de GeMAPS y su uso mediante OpenSMILE permitieron la resolución en la siguiente sección del subobjetivo 2.2-2.

A lo largo del capítulo 4 referente a los resultados, se realizó la comparativa directa entre todos los grupos de características y su desempeño en el modelo entrenado, completando el subobjetivo 2.2-2; mientras que además, gracias a que el modelo EMB es interpretable e indica la relevancia que han tenido las características, ha sido posible determinar cuáles de las mismas han resultado más útiles para realizar las clasificaciones de las voces en reales y falsas en relación al subobjetivo 2.2-3.

En este mismo capítulo y a lo largo del desarrollo en sus secciones de cada uno de los experimentos ejecutados, se ha ido comprobando gracias a distintas métricas (tales como el Accuracy, la Precisión o el F1 Score) la eficacia global del modelo cuando se entrenó únicamente con las características de GeMAPS (sección 4.2), o cuando se entrenaba con el añadido de las características obtenidas de la parte compleja de la señal (sección 4.3), o el de ambas parte compleja y parte real (sección 4.4) de forma que se completaba el subobjetivo 2.2-4, y con él, finalmente, el objetivo 2.2

Con los resultados obtenidos parece presentarse una rama de investigación interesante de cara a seguir estudiando hasta qué punto pueden ser relevantes las propiedades calculadas sobre la parte compleja y la parte real de la señal de audio y cuáles de estos atributos podrían llegar a marcar una diferencia a la hora de detectar ataques de spoofing de voz.

6

Trabajos futuros

- Al utilizar HABLA, se tiene la ventaja de trabajar con un dataset estandarizado y bastante limpio, donde de lo único que ha habido que hacer en cuanto al tratamiento de los datos ha sido de eliminar los silencios al inicio y al final del audio. Por lo tanto, se desconoce el desempeño de modelos que puedan usar estas características cuando en la muestra de audio exista ruido. Sería de interés comprobar la eficacia de las características calculadas en muestras que no estén limpias y que haya que procesar en el acto, principalmente porque a donde debe llegar la detección de deepfakes de audio es a poder hacerlo en tiempo real, por ejemplo en posibles llamadas donde seguramente exista ruido de fondo con el que haya que trabajar.

- En el caso de tener gran capacidad de cómputo, la continuación de estudio de este trabajo podría ser utilizar como base de características comparativas las provenientes de ComParE_2016 en vez de las de GeMAPS como se ha utilizado en esta ocasión. ComParE_2016 se centra en estudiar características relacionadas con los espectrogramas MEL, que como se indicó anteriormente tienen usualmente gran interés para el estudio de la voz. De esta forma, observar el desempeño de las características de la parte imaginaria junto con un tipo de características muy revelantes que evalúan aspectos diferentes puede proporcionar nuevas conclusiones.

- Otro camino a desarrollar sería calcular aquellas características presentadas en GeMAPS o ComParE_2016, siempre que tenga sentido, sobre la parte compleja de la señal de audio y hacer tal vez el mismo ejercicio que se ha desarrollado durante este proyecto para observar cómo afectan a la hora de

clasificar voces.

- Por último, una rama que, aunque no sea continuación de esta, está también íntimamente ligada es la investigación de si un modelo entrenado con la parte compleja de las señales de audio tendría mejor o peor desempeño. Esto sería aplicable tanto a modelos discriminativos como a modelos generativos. Aunque este trabajo se ha centrado en la parte de detectar los $deepfakes$, tal y como se dijo al comienzo, es un área de investigación que también trae consigo grandes beneficios, por lo que igualmente se anima a que se aplique esta investigación a ese área. Por ejemplo, se podría reentrenar algún modelo ya existente de generación de voces como Tortoise[35] [36] donde el input en vez de ser los espectrogramas usuales fueran aquellos que se generen de la parte compleja y contrastar posteriormente los resultados que se obtienen de este nuevo modelo en comparación al usual.

Bibliografía

[1] R. Hemavathi and R. Kumaraswamy, "Voice conversion spoofing detection by exploring artifacts estimates," *Multimedia Tools and Applications*, vol. 80, pp. 23 561–23 580, 6 2021.

[2] P. A. T. Flórez, R. Manrique, and B. P. Nunes, "Habla: A dataset of latin american spanish accents for voice anti-spoofing," vol. 2023-August, 2023.

[3] T. I. Contributors, "Interpretml," 2023, last accessed 6 July 2024. [Online]. Available: https://interpret.ml/docs/ebm.html

[4] J. Luque, "Voice cloning," 2023, last accessed 6 July 2024. [Online]. Available: https://colab.research.google.com/drive/1FCzltLTDw-QiU_Bv1qqdGlPOKGgp7tvO

[5] B. McFee, M. McVicar, D. Faronbi, and I. Roman, "librosa/librosa: 0.10.2.post1," *Zenodo*, 2024.

[6] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker, "Accurate intelligible models with pairwise interactions," vol. Part F128815, 2013.

[7] N. G. Barak, "Interpretml: Another way to explain your model," 2021. [Online]. Available: https://towardsdatascience.com/interpretml-another-way-to-explain-your-model-b7faf0a384f8

[8] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile - the munich versatile and fast open-source audio feature extractor," 2010.

[9] M. G. Martín, "Clasificación de voces a travÉs de series de fourier y redes neuronales." 2023.

[10] M. G. Martin, "ClasificaciÓn de voces a travÉs de series de fourier y redes neuronales," 2023. [Online]. Available: https://hdl.handle.net/10115/26283

[11] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, 1965.

[12] H. Jeon, Y. Jung, S. Lee, and Y. Jung, "Area-efficient short-time fourier transform processor for time–frequency analysis of non-stationary signals," *Applied Sciences*, vol. 10, p. 7208, 10 2020.

[13] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," vol. 2017-October, 2017.

[14] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, M. Kudinov, and J. Wei, "Diffusion-based voice conversion with fast maximum likelihood sampling scheme," 2022.

[15] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," 2018.

[16] Microsoft, "Azure text-to-speech," 2023, last accessed 6 July 2024. [Online]. Available: https://azure.microsoft.com/es-es/products/ai-services/text-to-speech

[17] MathWorks, "Espectrograma utilizando la transformada de fourier de tiempo corto." [Online]. Available: https://es.mathworks.com/help/signal/ref/spectrogram.html#bultmx7_sep_mw_c056db1e-cade-47af-bf56-37cd76eee5db

[18] B. Logan, "Mel frequency cepstral coefficients for music modeling," *International Symposium on Music Information Retrieval*, vol. 28, 2000.

[19] G. A. M. Mascorro and G. A. Torres, "Reconocimiento de voz basado en mfcc, sbc y espectrogramas," *Ingenius*, 2013.

[20] V. P. García, "Contrastes de normalidad," 2021, last accessed 6 July 2024.

[21] M. Molina, "Análisis de normalidad," 2022, last accessed 6 July 2024. [Online]. Available: https://anestesiar.org/2022/analisis-de-normalidad-una-imagen-vale-mas-que-mil-palabras/

[22] S. S. SHAPIRO and M. B. WILK, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, pp. 591–611, 12 1965.

[23] F. J. Massey, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, p. 68, 3 1951.

[24] N. Developers., "numpy.mean," 2024, last accessed 6 July 2024. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.mean.html#numpy-mean

[25] N. Developers, "numpy.nanpercentile," 2024, last accessed 6 July 2024. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.nanpercentile.html#numpy-nanpercentile

[26] N. Developers., "numpy.std," 2024, last accessed 6 July 2024. [Online]. Available: https://numpy.org/doc/stable/reference/generated/numpy.std.html#numpy-std

[27] F. Eyben, M. Wöllmer, and B. Schuller, "opensmile - the munich versatile and fast open-source audio feature extractor," *Proc. ACM Multimedia*, 2010.

[28] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. Andre, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan, and K. P. Truong, "The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing," *IEEE Transactions on Affective Computing*, vol. 7, pp. 190–202, 4 2016.

[29] M. F. Amin, "Confusion matrix in binary classification problems: A step-by-step tutorial," *Journal of Engineering Research*, vol. 6, 2022.

[30] ——, "Confusion matrix in three-class classification problems: A step-by-step tutorial," *Journal of Engineering Research*, vol. 7, 2023.

[31] H. Nori, S. Jenkins, P. Koch, and R. Caruana, "Interpretml: A unified framework for machine learning interpretability," 9 2019, last accessed 6 July 2024.

[32] P. Recuero De Los Santos, "Cómo interpretar la matriz de confusión," 2021, last accessed 6 July 2024. [Online]. Available: https://telefonicatech.com/blog/como-interpretar-la-matriz-de-confusion-ejemplo-practico

[33] F. Andino, M. Recarte, and M. Spilsbury, "La función delta de dirac," *REVISTA DE LA ESCUELA DE FÍSICA, UNAH*, vol. II, 6 2014.

[34] T. I. Contributors, "Faq," 2023, last accessed 10 July 2024. [Online]. Available: https://interpret.ml/docs/faq.html#

[35] J. Betker, "Better speech synthesis through scaling," 5 2023.

[36] ——, "Tortoise-tts," 2023, last accessed 6 July 2024. [Online]. Available: https://github.com/neonbjb/tortoise-tts

Apéndices

A

Gráficas individuales del desempeño de características

En esta sección se encuentran todas las gráficas generadas mediante InterpretML sobre el rendimiento y, en consecuencia, la relevancia de las características calculadas en base a la parte imaginaria y la parte real, según lo indicó el modelo EMB tras ser entrenado con esos datos.

A.1. Características del primer experimento

Recuérdese que en el primer experimento (sección 4.3) las características que se incluían eran las del dataset GeMAPS, junto con las calculadas de la parte imaginaria de la señal de audio: "ImagPartStd", "ImagPartMean", "ImagPartPercentile_5", "ImagPartPercentile_10", "ImagPartPercentile_50", "ImagPartPercentile_90" y "ImagPartPercentile_95".

Los resultados del experimento concluyeron que tanto "ImagPartStd" como "ImagPartMean" sí tenían relevancia al clasificar las voces en reales y falsas, mientra que el resto, que se corresponde a los percentiles, tenían una relevancia mucho menor. En las gráficas a continuación se podrá percibir de forma visual este resultado, ya que a menor perturbación sobre la línea menor efecto ha tenido sobre las clasificaciones, tal y como se explicaba en el apartado 4.1.2. Además hay que tener en cuenta la distribución, ya que por ejemplo en "ImagPartPercentile_50" se puede observar una pendiente decreciente, que podría indicarnos que ha sido relevante, pero si se observa la cantidad de ficheros que tienen esos valores es muy pequeña, por lo que a nivel global esta característica no ha resultado tan relevante.

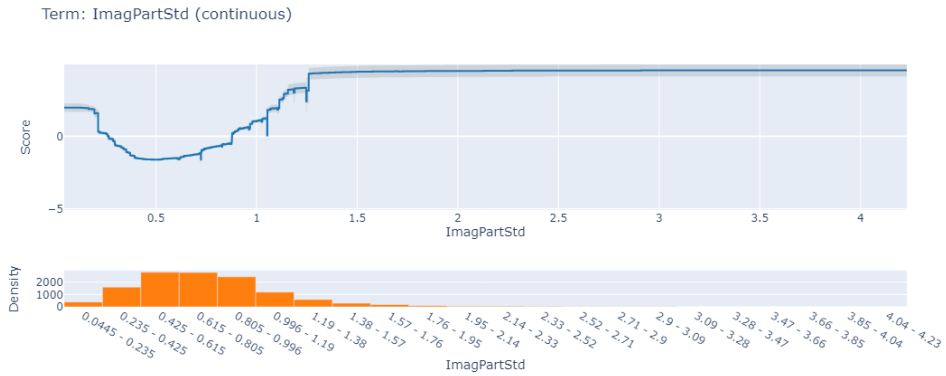


Figura A.1: Diagrama de desempeño de la característica "ImagPartStd" en el primer experimento.

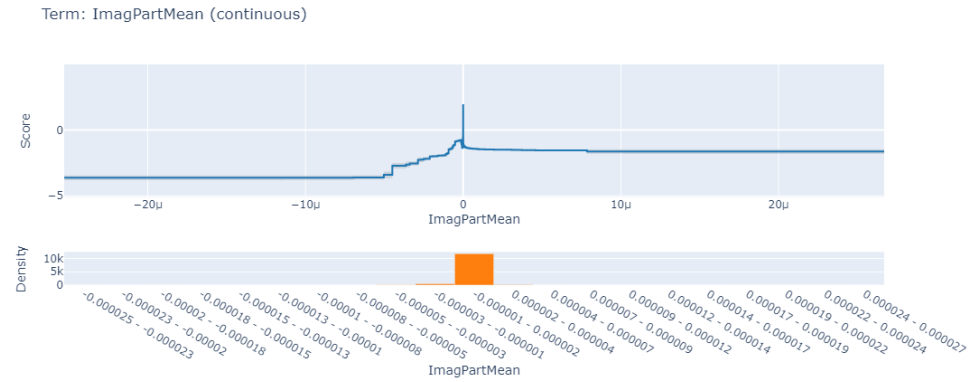


Figura A.2: Diagrama de desempeño de la característica "ImagPartMean" en el primer experimento.

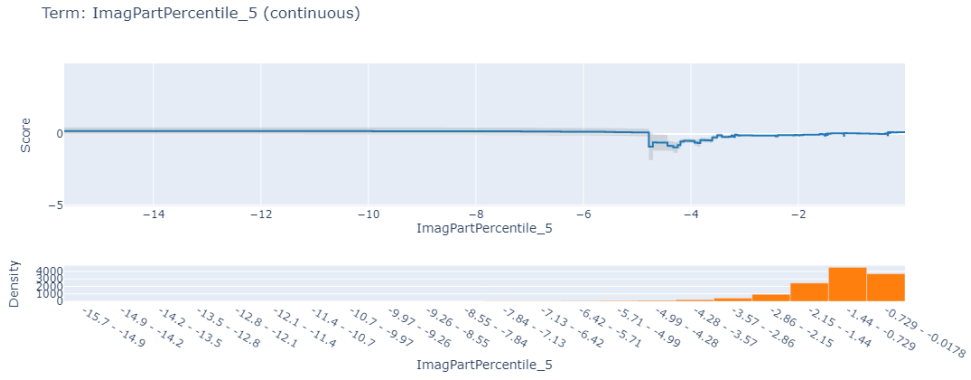


Figura A.3: Diagrama de desempeño de la característica "ImagPartPercentile_5" en el primer experimento.

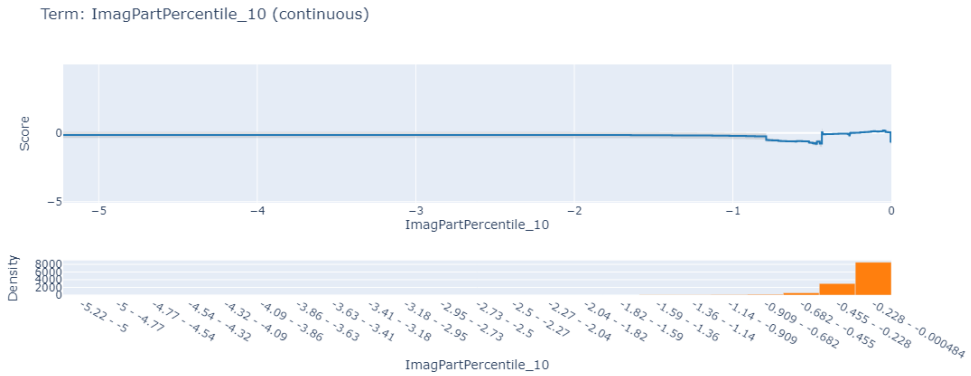


Figura A.4: Diagrama de desempeño de la característica "ImagPartPercentile_10" en el primer experimento.

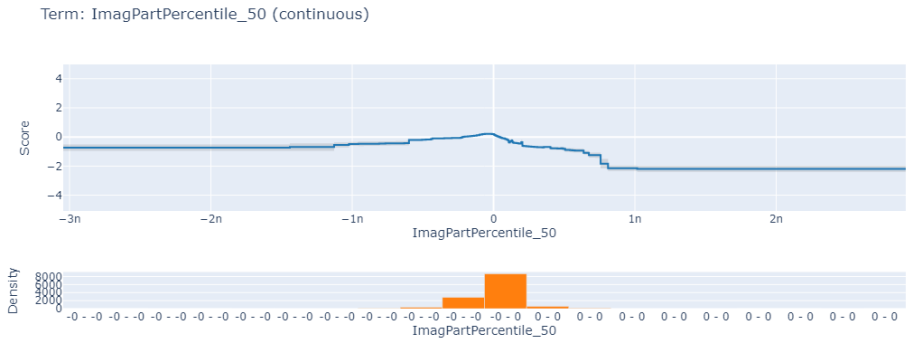


Figura A.5: Diagrama de desempeño de la característica "ImagPartPercentile_50" en el primer experimento.

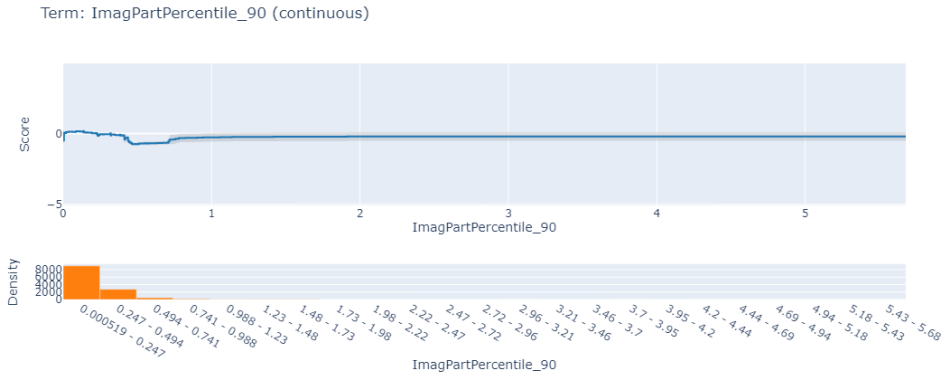


Figura A.6: Diagrama de desempeño de la característica "ImagPartPercentile_90" en el primer experimento.

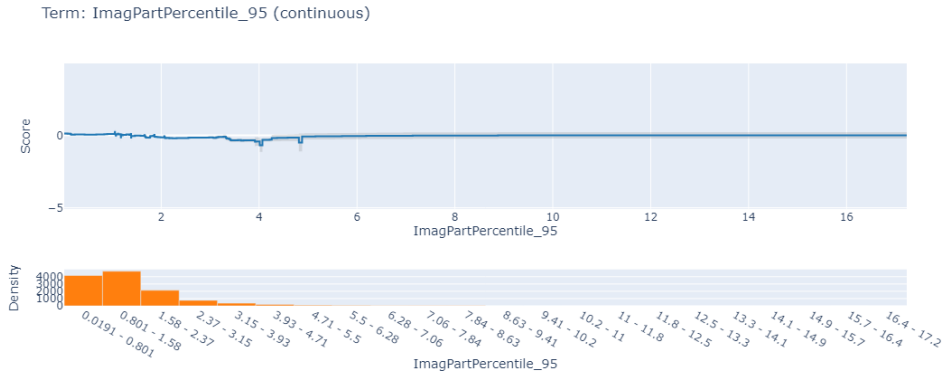


Figura A.7: Diagrama de desempeño de la característica "ImagPartPercentile_95" en el primer experimento.

A.2. Características del segundo experimento

En el segundo experimento (sección 4.4) las características incluidas fueron todas las que componían el dataset del primer experimento, las de GeMAPS y las calculadas sobre la parte compleja, y se añadieron las calculadas de la parte real de la señal de audio: "RealPartStd", "RealPartMean", "RealPartPercentile_5"; "RealPartPercentile_10", "RealPartPercentile_50", "RealPartPercentile_90" y "RealPartPercentile_95".

En este caso, resultaron que las partes que fueron importantes en el experimento anterior, "ImagPartStd" e "ImagPartMean", seguían siendo muy relevantes al clasificar las voces en reales y falsas, pero en un primer puesto apareció "RealPartMean" y el resto de características tuvieron una relevancia mucho menor. Se podrá comprobar el resultado de forma visual en las siguientes gráficas.

A.2.1. Características de la parte compleja

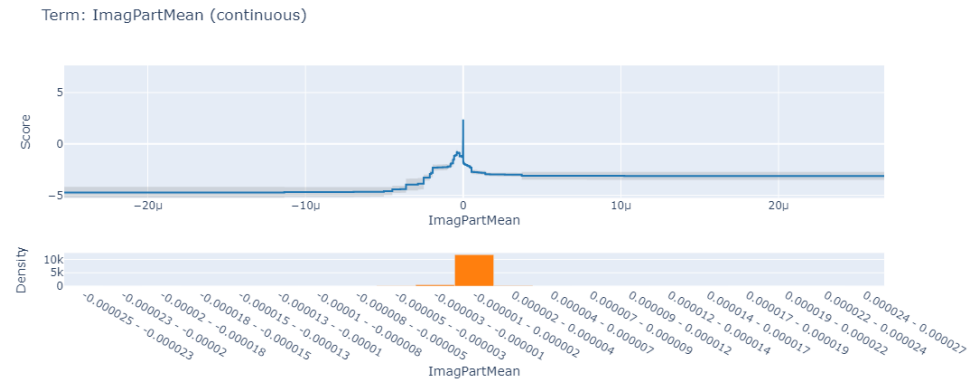


Figura A.8: Diagrama de desempeño de la característica "ImagPartMean" en el segundo experimento.

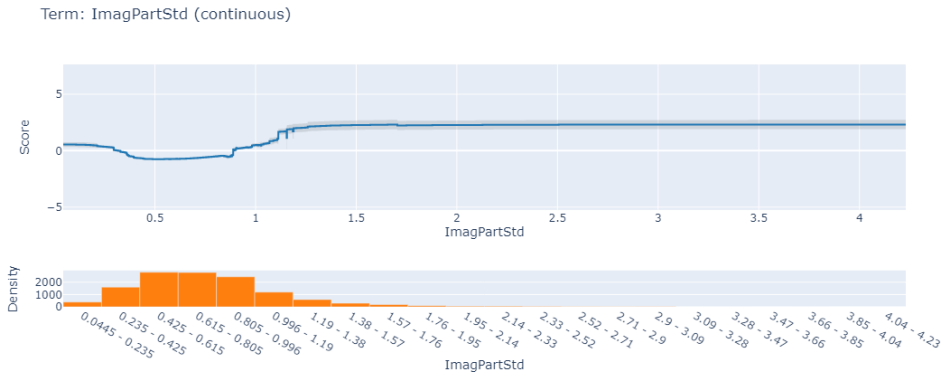


Figura A.9: Diagrama de desempeño de la característica "ImagPartStd" en el segundo experimento.

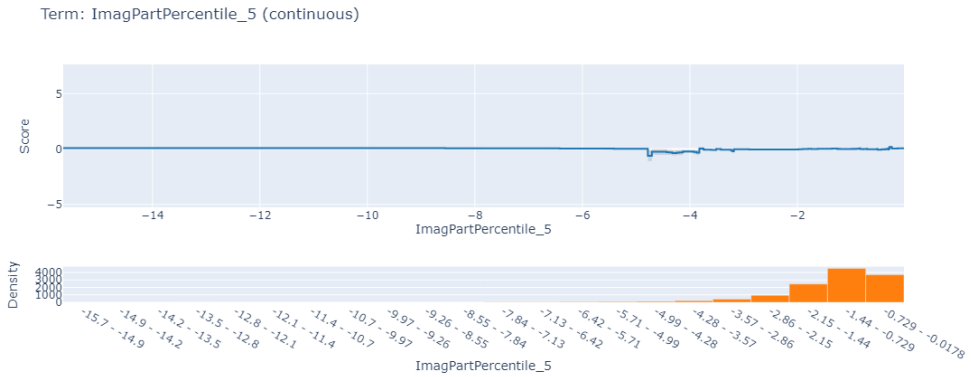


Figura A.10: Diagrama de desempeño de la característica "ImagPartPercentile_5" en el segundo experimento.

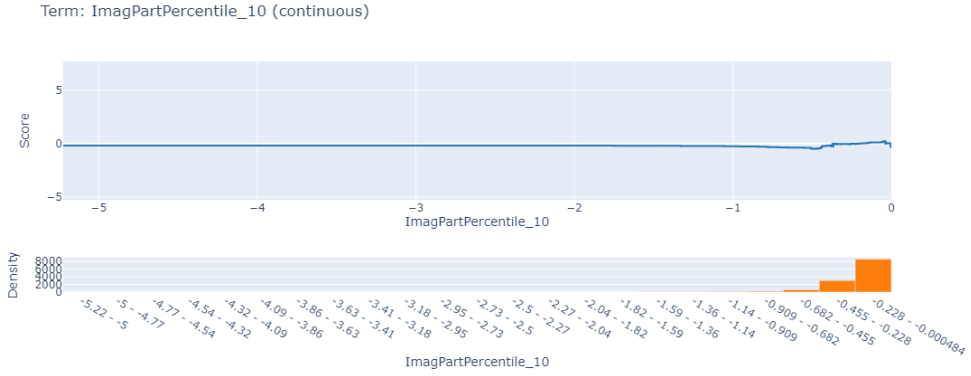


Figura A.11: Diagrama de desempeño de la característica "ImagPartPercentile_10" en el segundo experimento.

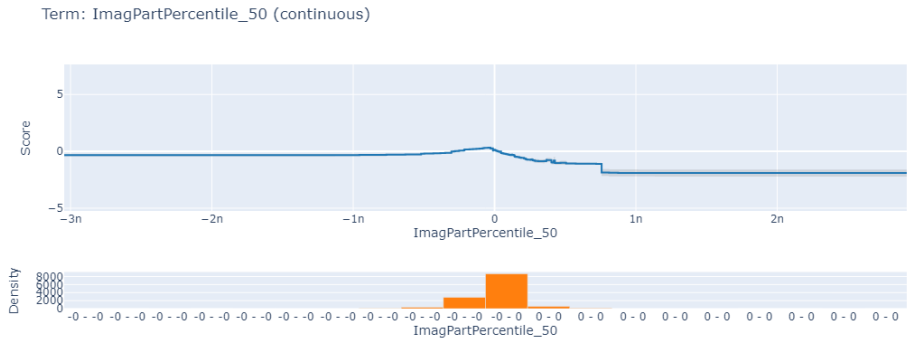


Figura A.12: Diagrama de desempeño de la característica "ImagPartPercentile_50" en el segundo experimento.

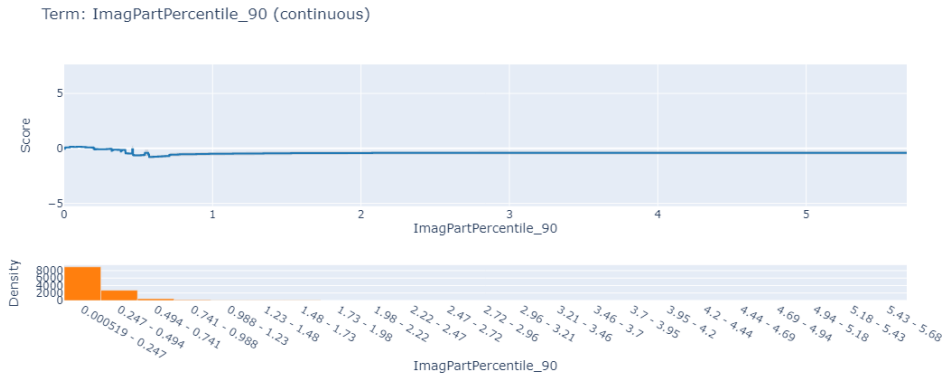


Figura A.13: Diagrama de desempeño de la característica "ImagPartPercentile_90" en el segundo experimento.

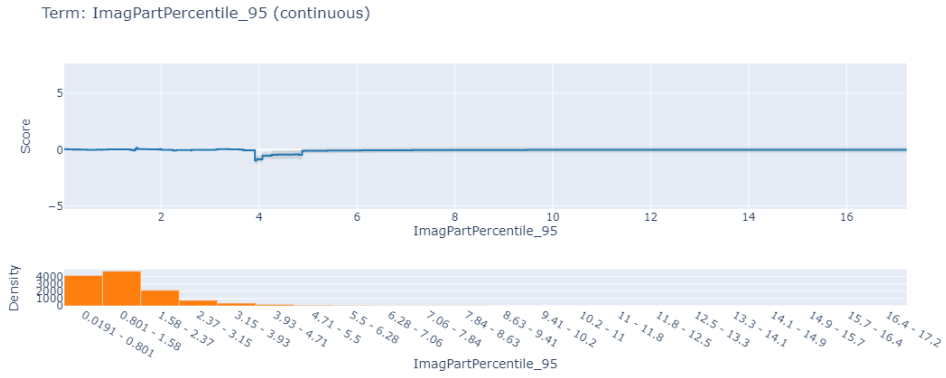


Figura A.14: Diagrama de desempeño de la característica "ImagPartPercentile_95" en el segundo experimento.

A.2.2. Características de la parte real

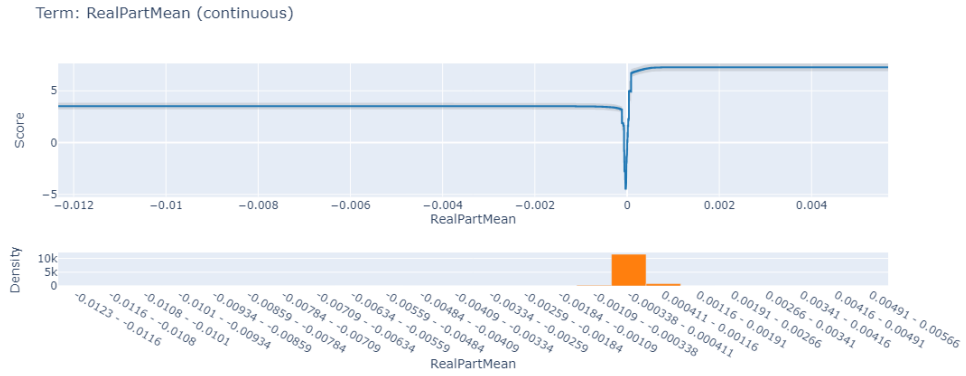


Figura A.15: Diagrama de desempeño de la característica “RealPartMean” en el segundo experimento.

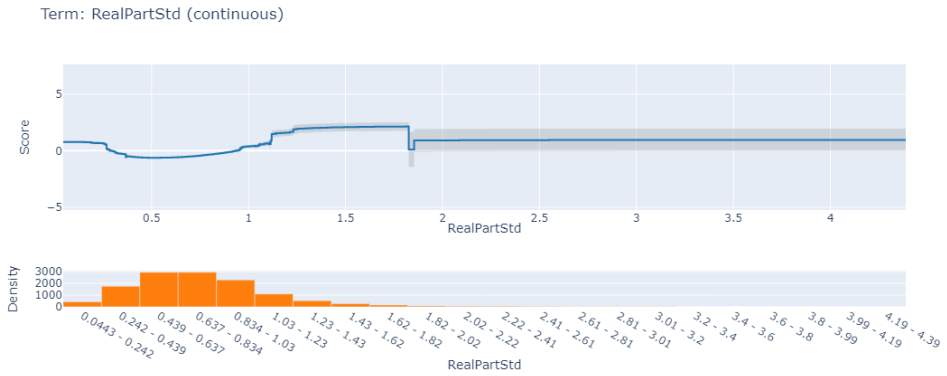


Figura A.16: Diagrama de desempeño de la característica “RealPartStd” en el segundo experimento.

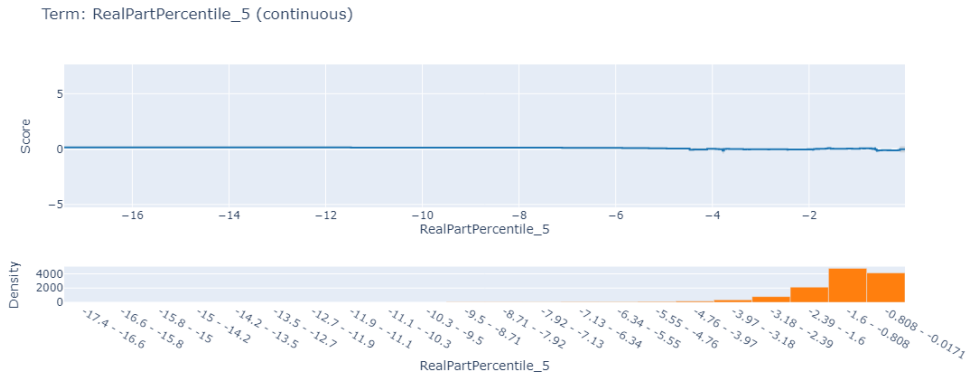


Figura A.17: Diagrama de desempeño de la característica "RealPartPercentile_5" en el segundo experimento.

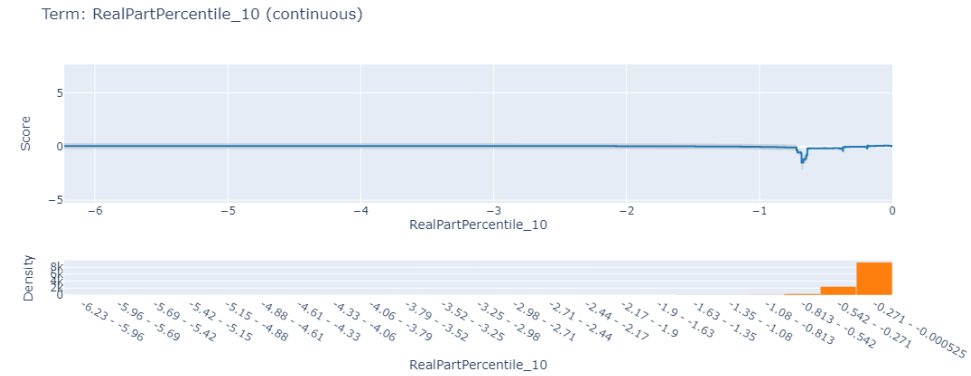


Figura A.18: Diagrama de desempeño de la característica "RealPartPercentile_10" en el segundo experimento.

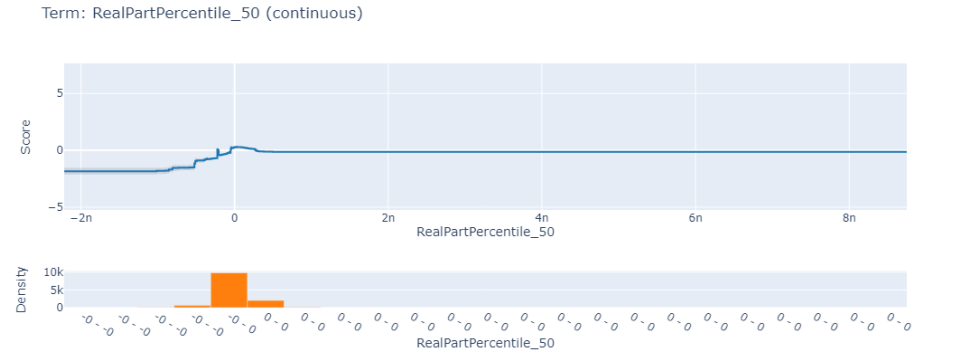


Figura A.19: Diagrama de desempeño de la característica "RealPartPercentile_50" en el segundo experimento.

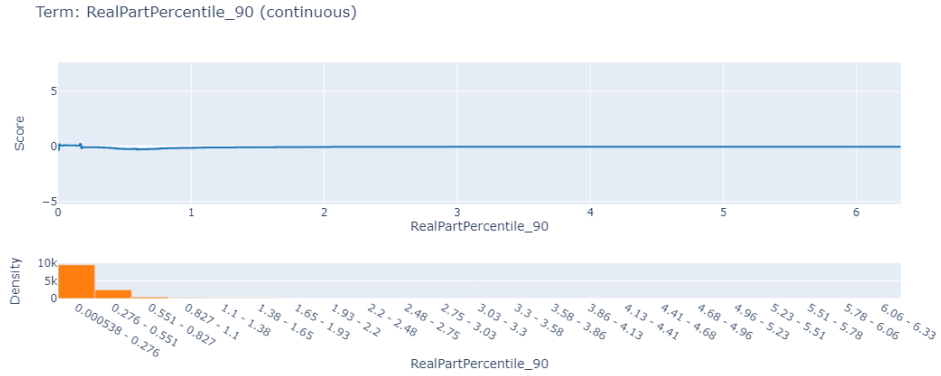


Figura A.20: Diagrama de desempeño de la característica “RealPartPercentile_90” en el segundo experimento.

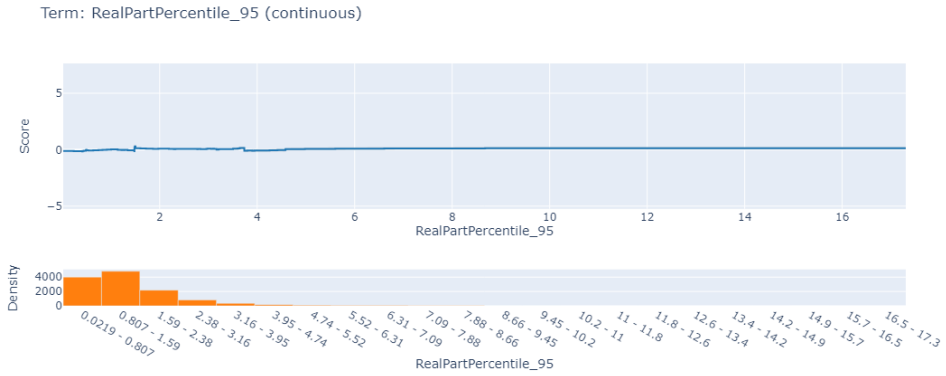


Figura A.21: Diagrama de desempeño de la característica “RealPartPercentile_95” en el segundo experimento.

A.3. Características del tercer experimento

En el tercer experimento (sección 4.5) las características que se incluían eran las del dataset GeMAPS, junto con las calculadas de la parte real de la señal de audio: "RealPartStd", "RealPartMean", "RealPartPercentile_5"; "RealPartPercentile_10", "RealPartPercentile_50", "RealPartPercentile_90" y "RealPartPercentile_95".

Los resultados del experimento indicaron que las características más relevantes para el modelo fueron "RealPartStd" y "RealPartMean", y en una menor medida, "RealPartPercentile_50". El resto de percentiles no resultaron apenas relevantes. A continuación se tienen las gráficas que muestran estos resultados.

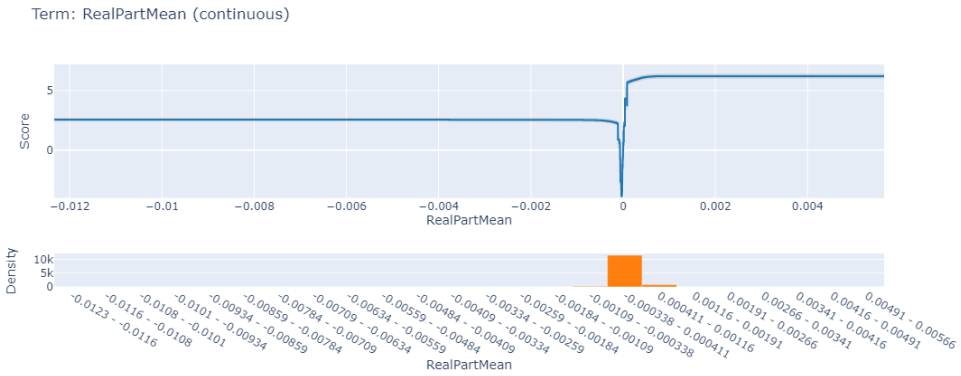


Figura A.22: Diagrama de desempeño de la característica "RealPartMean" en el tercer experimento.

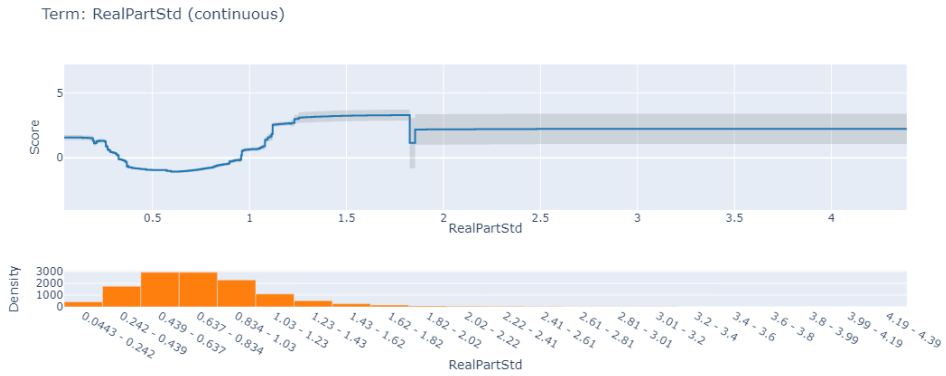


Figura A.23: Diagrama de desempeño de la característica "RealPartStd" en el tercer experimento.

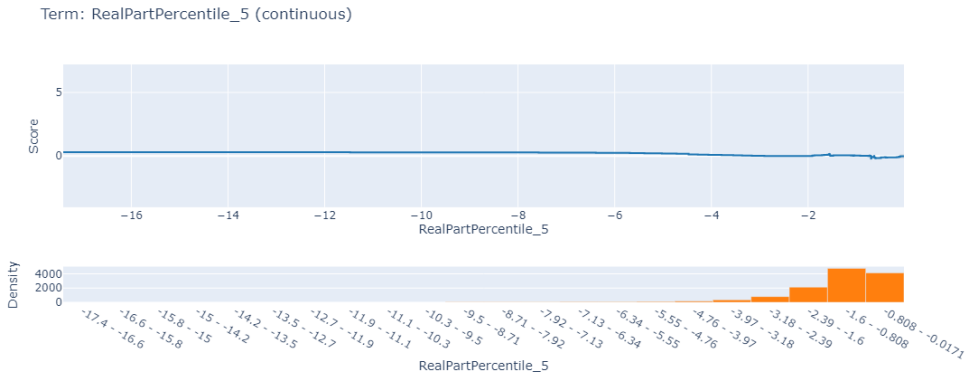


Figura A.24: Diagrama de desempeño de la característica "RealPartPercentile_5" en el tercer experimento.

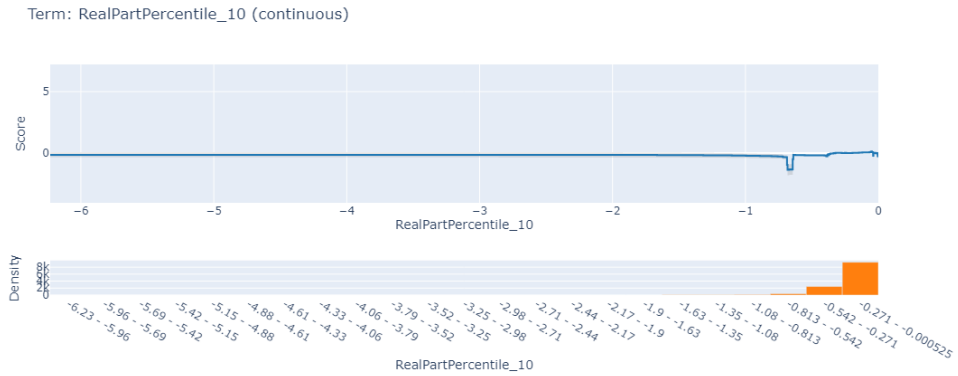


Figura A.25: Diagrama de desempeño de la característica "RealPartPercentile_10" en el tercer experimento.

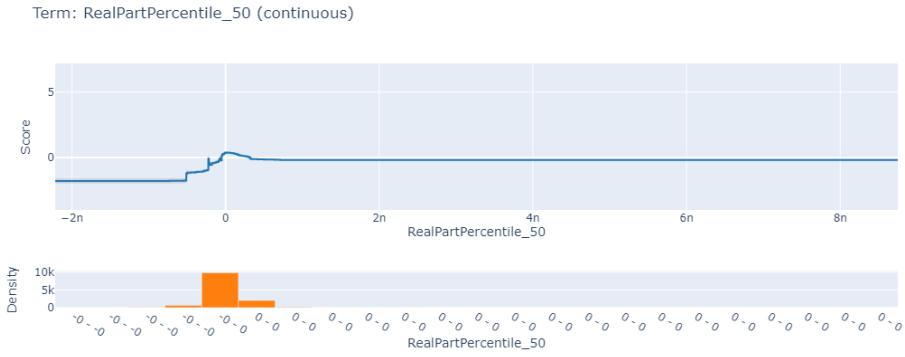


Figura A.26: Diagrama de desempeño de la característica "RealPartPercentile_50" en el tercer experimento.

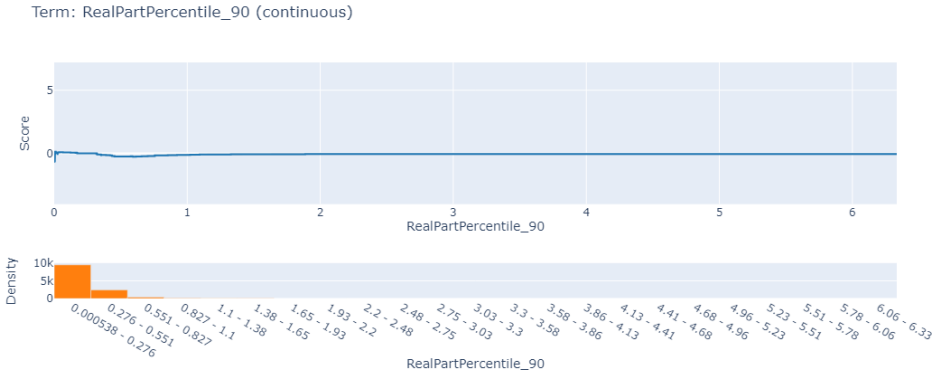


Figura A.27: Diagrama de desempeño de la característica "RealPartPercentile_90" en el tercer experimento.

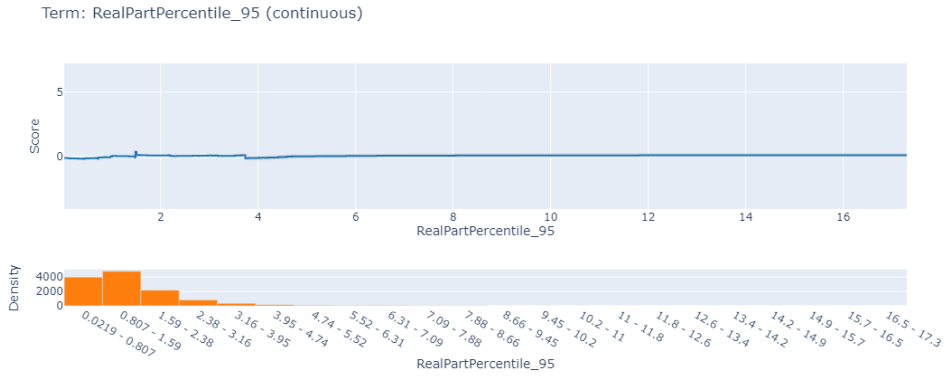


Figura A.28: Diagrama de desempeño de la característica "RealPartPercentile_95" en el tercer experimento.