PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
INGENIERÍA EN SISTEMAS Y COMPUTACIÓN



MODELO DE CLASIFICACIÓN PARA OBTENER LA DIRECCIÓN DE PROCEDENCIA DE UN SONIDO MEDIANTE ALGORITMOS DE MACHINE LEARNING

ANDERSON PAUL MEZA CANDO

Trabajo previo a la obtención del Título de Ingeniero en Sistemas y Computación

QUITO, JUNIO 2022

DEDICATORIA

Esta tesis está dedicada a:

A mis padres Carlos y Mariana quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir hoy un sueño y meta más, gracias por inculcar en mí el ejemplo de esfuerzo y responsabilidad. A mi abuelita Isaura que siempre ha estado pendiente de mí y de mis logros. A mi abuelito Manuel, que, aunque hoy no esté con nosotros le dedico todo mi esfuerzo con mi corazón por todo el apoyo y amor que un día me brindo, sé que estaría muy orgulloso.

Mi pareja Janina por su cariño, amor y apoyo incondicional durante todo este proceso, por estar conmigo en todo momento gracias. A toda mi familia porque con sus oraciones, consejos y palabras de aliento hicieron de mí una mejor persona y de una u otra forma me acompañan en todos mis sueños y metas.

Finalmente quiero dedicar este trabajo a mis compañeros especialmente a David. Por apoyarme cuando más los necesito, por extender su mano en momentos difíciles y por el cariño brindado cada día mil gracias.

AGRADECIMIENTO

Mi profundo agradecimiento a la Pontificia Universidad Católica del Ecuador, a toda la Facultad de Ingeniería, a mis profesores en especial al Ph.D. Henry Roa y Mgtr. Luis Oswaldo Espinosa quienes con la enseñanza de sus valiosos conocimientos hicieron que pueda crecer día a día como profesional, gracias a cada uno de ustedes por su paciencia, dedicación, apoyo incondicional y amistad.

Finalmente quiero expresar mi más grande y sincero agradecimiento al Dr. José Lucio, principal colaborador durante todo este proceso, quien con su dirección, conocimiento, enseñanza y colaboración permitió el desarrollo de este trabajo.

RESUMEN

En este trabajo se hizo uso de algoritmos de machine learning para entrenar un modelo con el propósito de obtener la dirección de procedencia de un sonido biauricular. Se aplicó algoritmos de aprendizaje supervisado, mismos que se entrenaron con audios de distintas procedencias. Los algoritmos utilizados fueron KN- Neighbors, Random Forest y Neural Network. Para el entrenamiento el dataset principal de audios se dividió en una dataset de entrenamiento 66% y un dataset de validación 34%. El modelo con mejor precisión luego del entrenamiento fue el de Neural Network que con un resultado de 77,7%. Sin embargo, en las cuatro pruebas con audios diferentes a los del entrenamiento, el modelo con mejor precisión fue el de KN-Neighbors con una precisión del 83,3%. Con esto se concluye que se generó un modelo con una precisión alta que cumple con el propósito del proyecto.

INDICE GENERAL

1. INTRODUCCIÓN 1

1.1. JUSTIFICACIÓN 1

1.2. PLANTEAMIENTO DEL PROBLEMA 2

1.3. OBJETIVOS 3

2. MARCO TEÓRICO 4

2.1. ANTECEDENTES 4

2.2. INDICIOS ACÚSTICOS PARA IDENTIFICACIÓN DE LA PROCEDENCIA DEL SONIDO. 6

2.3. AURILIZACIÓN Y SÍNTESIS BIAURICULAR 13

2.4. APRENDIZAJE AUTOMÁTICO SUPERVISADO Y NO SUPERVISADO 16

2.5. TÉCNICAS CLASIFICACIÓN MACHINE LEARNING 20

3. METODOLOGÍA Y MATERIALES 29

3.1. DEFINICIÓN DEL DATASET DE SEÑALES DIRECCIONALES 29

3.2. IDENTIFICACIÓN DE CARACTERÍSTICAS PRINCIPALES PARA IDENTIFICACIÓN DE LA POSICIÓN DE LA FUENTE SONORA. 45

3.3. DEFINICIÓN DEL MÉTODO DE MACHINE LEARNING ÓPTIMO PARA CLASIFICACIÓN DE ESTE TIPO DE DATOS. 66

3.4. DEFINICIÓN DE LA HERRAMIENTA DE PROCESAMIENTO ACÚSTICO Y MACHINE LEARNING 68
3.5. LIBRERÍAS Y CLASES DE LA HERRAMIENTA PARA PROCESAMIENTO ACÚSTICO Y PARA CLASIFICACIÓN DE SEÑALES 71
3.6. REQUERIMIENTOS FUNCIONALES 72
4. IMPLEMENTACIÓN DEL MODELO 73
4.1. DEFINICIÓN DE LA ARQUITECTURA DEL MODELO 73
4.2. ENTRENAMIENTO DEL MODELO 83
4.3. PRUEBAS DE VALIDACIÓN DEL ENTRENAMIENTO 86
5. RESULTADOS 95
5.1. PRUEBAS DE VALIDACIÓN DEL MODELO 95
5.2. DISCUSIÓN DE RESULTADOS 125
6. CONCLUSIONES Y RECOMENDACIONES 127
7. BIBLIOGRAFÍA 129
8. ANEXOS 131

ÍNDICE DE FIGURAS

Figura 1 1
Cono de confusión 1
Figura 2 8
Localización del sonido 8
Figura 3 9
Planos relativos al oyente y ángulos de posicionamiento de la fuente 9
Figura 4 10
Respuestas a impulso registradas con una cabeza artificial para □ = 30° y □ = 0° 10
Figura 5 11
Cono de confusión 11
Figura 6 12
Indicio aportado por la rotación de la cabeza 12
Figura 7 13
Percepción simétrica respecto al eje Interaural 13
Figura 8 15
Síntesis biauricular 15
Figura 9 16
Esquema de aplicación del Modelo de cabeza 16
Figura 10 19
Esquema de aplicación del Modelo de cabeza 19
Figura 11 20
Diagrama de bloques de aprendizaje no supervisado 20
Figura 12 22
Clasificación de vecinos más cercanos 22
Figura 13 24
Estructura de un árbol de decisión 24
Figura 14 26
Modelo genérico de una neurona artificial 26
Figura 15 27
Funciones de activación 27
Figura 16 28
Red neuronal monocapa 28
Figura 17 28
Red Neuronal multicapa 28
Figura 18 29
Representación de la esfera formada por los ejes de los ángulos de elevación y azimut 29
Figura 19 31
Muestras de los diferentes ángulos de elevación 31
Figura 20 32
Muestras de los diferentes ángulos de elevación 32
Figura 21 33
Contenido de la carpeta "elev0" 33
Figura 22 34
Gráfico de la onda del sonido de un timbre 34

Figura 23 35
Icosaedro 35
Figura 24 36
Icosaedro abierto 36
Figura 25 36
Imagen de los vectores resultado de la función createRays ( ) 36
Figura 26 37
Gráfico de la esfera con radio=1 unidad 37
Figura 27 38
Vista frontal de la esfera 38
Figura 28 39
Esfera generada por puntos 39
Figura 29 40
Onda del audio mono auricular base 40
Figura 30 41
Onda del audio izquierdo 41
Figura 31 42
Onda del audio derecho 42
Figura 32 43
Ondas Izquierda y derecha 43
Figura 33 44
Repositorio de audios 44
Figura 34 49
Vista frontal de la esfera formada por los ángulos de elevación y azimut 49
Figura 35 50
Vista superior de la esfera formada por los ángulos de elevación y azimut 50
Figura 36 51
Imagen de la esfera "abierta" 51
Figura 37 58
Ondas auricular izquierdo y auricular derecho 58
Figura 38 61
Correlación entre los auriculares derecho e izquierdo 61
Figura 39 63
Ondas auriculares izquierdo y derecho 63
Figura 40 63
Correlación cruzada 64
Figura 41 67
Características de la zona 67
Figura 42 73
Formato de archivo CSV 73
Figura 43 74
Muestra de los primeros 22 registros del dataset 74
Figura 44 74
Herramienta de Orange 74
Figura 45 75
Gráfico de dispersión 75
Figura 46 76

Gráfico de dispersión 76
Figura 47 77
Muestra de los valores atípicos 77
Figura 48 78
Valores de la correlación 78
Figura 49 79
Valores de la correlación 79
Figura 50 80
Muestra de los límites 80
Figura 51 81
Muestra de los límites 81
Figura 52 82
Gráfico de dispersión 82
Figura 53 83
Arquitectura del entrenamiento del modelo en Orange 83
Figura 54 85
Imagen flujo del entrenamiento con los tres modelos principales en la herramienta Orange 85
Figura 55 87
Resultados entrenamiento KNN 87
Figura 56 88
Matriz de confusión 88
Figura 57 88
Matriz de confusión en porcentajes 88
Figura 58 89
Resultados entrenamiento Random Forest 89
Figura 59 90
Matriz de confusión en porcentaje resultado del entrenamiento Random Forest con 100 árboles 90
Figura 60 90
Matriz de confusión en porcentaje resultado del entrenamiento Random Forest con 100 árboles 90
Figura 61 92
Resultados entrenamiento Red Neuronal 92
Figura 62 93
Matriz de confusión resultado del entrenamiento de Red Neuronal con dos capas de 2000 neuronas 93
Figura 63 93
Matriz de confusión en porcentaje resultado del entrenamiento de Red Neuronal con dos capas de 2000 neuronas 93
Figura 64 96
Gráfico de la onda del sonido de prueba 'audio2.wav' 96
Figura 65 96
Tabla de resultado de las pruebas 96
Figura 66 97
Matriz de confusión resultado de las pruebas con Red Neuronal 97
Figura 67 97
Matriz de confusión en porcentaje resultado de las pruebas con Red Neuronal 97
Figura 68 98
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Red Neuronal 98
Figura 69 99

Matriz de confusión resultado de las pruebas con Random Forest 99
Figura 70 99
Matriz de confusión en porcentaje resultado de las pruebas con Random Forest 99
Figura 71 100
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Random Forest 100
Figura 72 101
Matriz de confusión resultado de las pruebas con KNN 101
Figura 73 101
Matriz de confusión en porcentaje resultado de las pruebas con KNN 101
Figura 74 102
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de KNN 102
Figura 75 103
Gráfico de la onda del sonido de prueba 'audio3.wav' 103
Figura 76 103
Tabla de resultado de las pruebas 103
Figura 77 104
Matriz de confusión resultado de las pruebas con Red Neuronal 104
Figura 78 104
Matriz de confusión en porcentaje resultado de las pruebas con Red Neuronal 104
Figura 79 105
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Red Neuronal 105
Figura 80 106
Matriz de confusión resultado de las pruebas con Random Forest 106
Figura 81 106
Matriz de confusión en porcentaje resultado de las pruebas con Random Forest 106
Figura 82 107
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Random Forest 107
Figura 83 108
Matriz de confusión resultado de las pruebas con KNN 108
Figura 84 108
Matriz de confusión en porcentaje resultado de las pruebas con KNN 108
Figura 85 109
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de KNN 109
Figura 86 110
Gráfico de la onda del sonido de prueba 'audio4.wav' 110
Figura 87 110
Tabla de resultados de las pruebas 110
Figura 88 111
Matriz de confusión resultado de las pruebas con Red Neuronal 111
Figura 89 111
Matriz de confusión en porcentaje resultado de las pruebas con Red Neuronal 111
Figura 90 112
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Red Neuronal 112
Figura 91 113
Matriz de confusión resultado de las pruebas con Random Forest 113
Figura 92 113
Matriz de confusión resultado de las pruebas con Random Forest 113

Figura 93 114
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Random Forest 114
Figura 94 115
Matriz de confusión resultado de las pruebas con KNN 115
Figura 95 115
Matriz de confusión resultado de las pruebas con KNN 115
Figura 96 116
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de KNN 116
Figura 97 117
Gráfico de la onda del sonido de prueba 'audio5.wav' 117
Figura 98 117
Tabla de resultado de las pruebas 117
Figura 99 118
Matriz de confusión resultado de las pruebas con Red Neuronal 118
Figura 100 118
Matriz de confusión en porcentaje resultado de las pruebas con Red Neuronal 118
Figura 101 119
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Red Neuronal 119
Figura 102 120
Matriz de confusión resultado de las pruebas con Random Forest 120
Figura 103 120
Matriz de confusión en porcentaje resultado de las pruebas con Random Forest 120
Figura 104 121
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Random Forest 121
Figura 105 122
Matriz de confusión resultado de las pruebas con KNN 122
Figura 106 122
Matriz de confusión en porcentaje resultado de las pruebas con KNN 122
Figura 107 123
Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de KNN 123
Figura 108 124
Tabla de resultado de las pruebas con los modelos 124

1. INTRODUCCIÓN

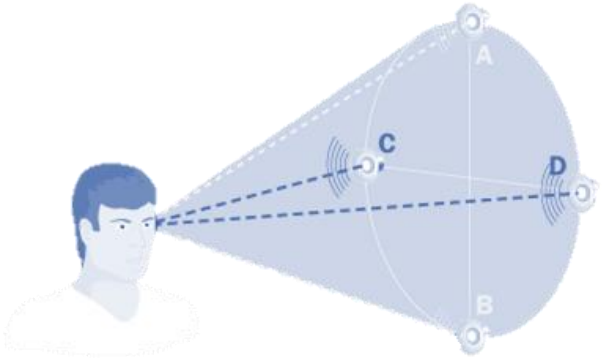
1.1. JUSTIFICACIÓN

Este proyecto busca plantear un modelo basado en algoritmos de Machine Learning para reconocer la procedencia de un sonido. Para esto, se identificarán las características que hacen que un sonido tenga el efecto de espacialidad a partir de un dataset de sonidos direccionales previamente generado, y con tales características se entrenará un modelo de clasificación.

Es relevante hacer hincapié que el modelo recibirá como entrada muestras de dos señales con un sonido estéreo y devolverá la dirección de procedencia de la fuente que lo produce. Esto tratando de emular la capacidad natural humana para identificar la posición de una fuente sonora, tomando en cuenta algunas de las dificultades que se suelen presentar, por ejemplo: los conos de confusión ilustrados en la figura 1. Esta figura ilustra una zona donde las diferencias interauriculares son constantes (distancias entre cualquier punto ubicado en su superficie, y los puntos que representan a los oídos), por lo que existe una confusión en la ubicación de la fuente sonora. (Cetta, P. C., 2007).

Figura 1

Cono de confusión



Fuente: Realización Propia (2021)

El aspecto relacionado con el uso de modelos de machine learning hacen que este abordaje sea novedoso. Esto porque, la revisión bibliográfica preliminar realizada indica que sería un trabajo innovador en el área.

Las aplicaciones que se pueden lograr con este modelo pueden darse en dispositivos de navegación asistida electrónicamente para personas no videntes. Por lo cual, una alternativa es usar audífonos ecualizados que reproduzcan audio 3D capaz de estimular la capacidad natural de identificación de la posición de la fuente sonora (obstáculo) de la persona con discapacidad visual. En este caso, el modelo podría ofrecer señales correctivas mejorando el desempeño del dispositivo ETA (Electronic Travel Aid). Otra aplicación puede ser el uso de este modelo para perfeccionar la inmersión acústica en videojuegos en tercera dimensión mejorando el efecto de realidad virtual.

1.2. PLANTEAMIENTO DEL PROBLEMA

El problema que se genera es por la subjetividad de las personas para localizar de forma exacta la procedencia de un sonido y también porque existen ciertas posiciones donde la localización es más complicada que en otras. Por tal motivo, en este proyecto se buscará perfeccionar la percepción direccional sonora tratando de evaluar, a través de mecanismos de Machine Learning, las diferencias en las características objetivas de un mismo sonido pero que proviene de diferentes direcciones.

Una de las mayores aplicaciones de los algoritmos de Machine Learning es la identificación de patrones. En temas relacionados con la percepción sensorial, los resultados dependen de la percepción subjetiva de individuos que no siempre es generalizable en un grupo numeroso de personas.

La herramienta que se usará en principio será un computador en donde se generará el dataset de entrenamiento (con técnicas convencionales de procesamiento de señales), se entrenará y probará el modelo. A demás se generarán otros audios direccionales para probar el modelo con datos diferentes a los que se usaron en el proceso de entrenamiento.

En función de esta problemática se plantea la siguiente pregunta principal de investigación:

• ¿Cuáles son las principales características que se debe tomar en cuenta para conocer la dirección de procedencia de un sonido?

Y las siguientes preguntas secundarias:

• ¿De qué forma es posible perfeccionar la percepción direccional de procedencia de un sonido?

• ¿Qué patrones se pueden encontrar en una señal de sonido?

1.3.OBJETIVOS

Objetivo General

- Desarrollar un modelo que permita el reconocimiento de la dirección de procedencia de un sonido mediante algoritmos de clasificación de Machine Learning.

Objetivos Específicos

- Analizar, y escoger el algoritmo de clasificación apropiado para el entrenamiento y optimización del modelo.
- Desarrollar un Dataset para poder entrenar el modelo de Machine Learning.
- Entrenar y optimizar el modelo.
- Probar el funcionamiento del modelo con audios diferentes que no formaron parte del entrenamiento del modelo.

2. MARCO TEÓRICO

2.1. ANTECEDENTES

Las aplicaciones a partir de una tecnología que permita la localización de sonido son múltiples, sin embargo, la mayoría de los proyectos que van direccionados en este tema no cuentan con algoritmos de machine learning. El proyecto que destaca en el uso de machine learning para encontrar la dirección de la fuente de un sonido es el proyecto SoundWater de la Asociación Española de Abastecimientos de Agua y Saneamiento (AEAS) el cual resuelve el problema de localizar las fugas en un terreno en las redes de distribución de agua potable, por lo que se escucha el sonido captado en diferentes puntos de la red hasta que encuentran la fuga o descartan que el sonido esté relacionado con otro evento, se extrae la dirección de donde proviene la fuga marcando así el área y se envía el equipo de reparaciones. (Cardelus & Lorenzo, 2019)

También se encuentran proyectos que usan algoritmos de clasificación mediante machine learning que no se enfocan directamente a clasificar la dirección de la fuente de un sonido, sino que realizan una clasificación enfocada al reconocimiento de la fuente o el significado del sonido. Como ejemplo de esto tenemos el proyecto de clasificación de los sonidos cardíacos usando ondículas y redes neuronales en donde por medio de los sonidos cardíacos en un examen clínico permite determinar si un paciente debe ser referido a un especialista. El fono cardiograma (PCG), por sus siglas en inglés, corresponde al registro de estos sonidos. El objetivo de este trabajo es la evaluación de un esquema fundamentado en dos algoritmos propuestos durante el desafío PhysioNet 2016, el primero basado en ondículas y el segundo en una red neuronal convolucional (RNC), para evaluar el desempeño en la clasificación de los sonidos cardíacos (normal/anormal). (Peralta, Carrión, Tenesaca, & Vázquez Rodas, 2017)

A partir de aquí se encuentran proyectos que se enfocan a la localización de los sonidos mediante técnicas matemáticas, uso de dispositivos o con otro tipo de algoritmos de forma tradicional. “la forma de generar algoritmos que interpretaran datos era la siguiente: 1. Una persona escribía el algoritmo y se lo transmitía a la máquina en forma de código. 2. La máquina se limitaba a leer unos datos y ejecutar el algoritmo. 3. De esta forma, se generaban las predicciones” (Fraga Domingo, 2019). Es este caso tenemos el proyecto de Localización del sonido utilizando beamforming y su representación en un dispositivo de realidad mixta, aquí se utiliza la técnica de formación de haces “Sum-and-Delay” que genera un ángulo aproximado de su ubicación mientras reduce el ruido en la señal, es popular por su versatilidad ya que funciona como tipo de filtro espaciotemporal. Es decir, manipula las señales en el dominio del tiempo de llegada del sonido. Aquí se logró aproximar la exactitud con un error medio del 2,5% en distancias de hasta 3 metros largo. (Valencia Palma, 2019). Existe otro proyecto que se enfoca a la localización de sonidos en entorno abierto por personas ciegas, aquí de igual manera tampoco se usa ninguna técnica de machine learning para lograr su objetivo de que las personas invidentes sean capaces de localizar objetos mediante sonidos simulados virtualmente y detectarlos con gran precisión, en cambio se utiliza un dispositivo sensor que está basado un láser que emite una luz láser con una definición de 64 puntos con una diferencia de un grado de separación ente cada punto. (Lengua, Dunai, Peris-Fajarnés, & Brusola, 2013)

Gracias a los proyectos anteriores se sabe que el uso de machine learning dentro de la localización de fuentes sonoras es poco común, en cambio existe gran uso en cuanto a clasificación de tipos de sonidos, por lo que se observa un gran potencial en este proyecto que a partir del modelo que se desarrolle podrá ser tomado para varias aplicaciones que necesiten saber la dirección de procedencia de un sonido.

2.2. INDICIOS ACÚSTICOS PARA IDENTIFICACIÓN DE LA PROCEDENCIA DEL SONIDO.

El sonido se transmite a través de vibraciones de las partículas del medio donde actúa en forma de ondas mecánicas que se propagan en dicho medio. Por lo tanto, es una energía mecánica procedente de una superficie en vibración y se transmite por series cíclicas de compresiones y enrarecimientos de las moléculas de los materiales que atraviesa. Se puede transmitir a través de los gases, líquidos y sólidos. El sonido consiste en el movimiento de una onda que se presenta cuando una fuente de sonido pone en movimiento las partículas más cercanas del aire, que se propagan de forma uniforme en todas direcciones, disminuyendo su amplitud a medida que se alejan de la fuente. El movimiento gradualmente se extiende a las partículas de aire más lejanas de su fuente de emisión. (Lazcano Enciso , 2006)

Un sonido se caracteriza por su intensidad y tono:

INTENSIDAD: Es la cantidad de energía que se transmite al oído y está relacionada con la amplitud de la onda sonora; esta amplitud puede expresarse en función del cambio de presión máximo o bien de la raíz cuadrada de la presión media en el tímpano, pero es bastante complejo de medir y por ello se utiliza una escala relativa: la de decibeles o decibelios (dB). Un decibel representa un aumento de intensidad real de 1.26 veces. A mayor amplitud, mayor intensidad. El valor de 0 decibeles es el que se encuentra en el umbral auditivo promedio de las personas. (Costa, J; Madrid, J A& Zamora, S, 1993).

TONO: Es la percepción psicofisiológica de la frecuencia. La frecuencia es el número de ondas por unidad de tiempo y se mide en Hercios (Hz). A mayor frecuencia, vías y\o centros superiores. (Costa, J., Madrid, J. A., & Zamora, S. (Eds.). 1993).

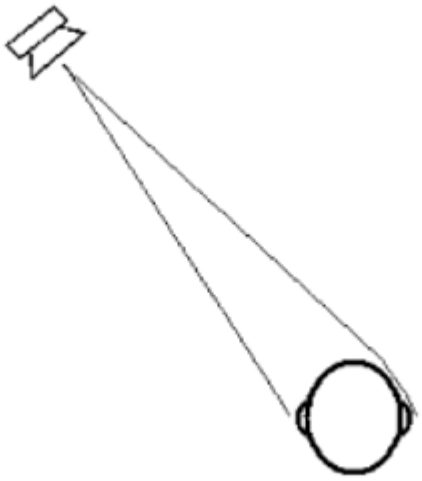
En el ser humano la audición es un sentido mecanorreceptor, “el oído responde a la vibración mecánica de las ondas sonoras en el aire. Estas ondas que llegan al oído viajan por el conducto auditivo externo hasta llegar al tímpano. Al comprimir y descomprimir el aire, mueven la membrana hacia dentro y hacia fuera. Este movimiento es transmitido por la cadena mecánica de huesecillos hasta la ventana oval o conducción aérea, la cual pone en movimiento la endolinfa contenida en el caracol. El movimiento vibratorio de la endolinfa hace oscilar la membrana basilar, en donde están ubicadas las células ciliadas. Estos cilios están fijos en su extremo distal embebidos por la matriz gelatinosa de la membrana tectorial. Puesto que el punto de apoyo de ambas membranas en el hueso temporal es distinto, la oscilación de la membrana basilar somete a un desplazamiento por cizalladura a los cilios. También se puede mover la endofilia como consecuencia de las vibraciones de las paredes óseas del caracol (conducción ósea)”. (Costa, J., Madrid, J. A., & Zamora, S. (Eds.). 1993)

Audición Biauricular

El sonido viaja a través del entorno y obstáculos. Gracias al hecho de que los seres humanos tenemos dos oídos (audición biauricular), lo que permite tener dos señales que por lo general presentan un diferente nivel y retardo Figura 2. El cerebro tiene la habilidad de comparar ambas señales y así poder localizar sonidos. (Jaramillo, 2007)

Figura 2

Localización del sonido



Fuente: Jaramillo (2007).

Si una fuente sonora se encuentra en el plano horizontal respecto a la cabeza, el ser humano pude localizar la dirección de procedencia. Esto alrededor de un grado de precisión. El cerebro hace uso de dos características de las señales: (Jaramillo, 2007)

Interaural Time Difference – ITD (Diferencia de Tiempo Interauricular en español)

Esta es la diferencia de tiempo y de las señales que llegan a ambos oídos. Los sonidos tienen un retardo diferente en cada oído debido al recorrido del sonido y la posición de la fuente. Si se produce un sonido desde el lado derecho de la cabeza, el oído derecho captará esta señal primero y viceversa. (Jaramillo, 2007)

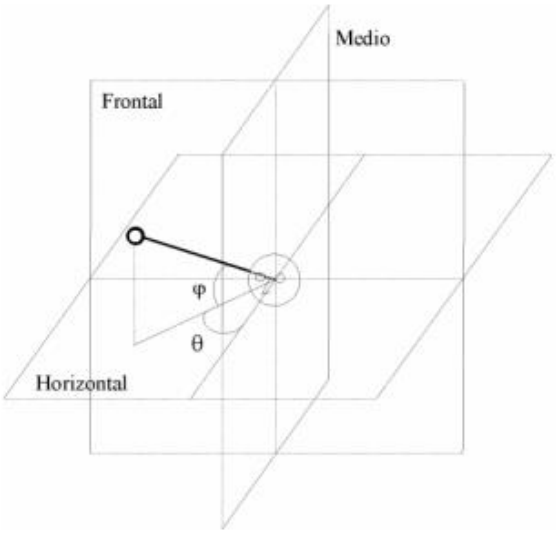
Interaural Level Difference – ILD (Diferencia de Nivel Interauricular en español)

Es la diferencia de nivel en que las señales de sonido llegan a ambos oídos. Las señales llegan con una diferencia de nivel en cada oído. Debido al recorrido del sonido y los obstáculos en el camino. Mientras más distancia recorra la onda, la energía será menor. La señal que llegue con un nivel de energía mayor es la que menos recorrido hizo. (Jaramillo, 2007)

Para ubicar un evento sonoro en el espacio se usa coordenadas esféricas mediante dos ángulos. Uno medido sobre el plano horizontal – ángulo de azimut (θ), para el que consideramos 0° al frente- y otro que se eleva sobre este plano y mide la pendiente del vector con origen en el centro de la cabeza del oyente, y extremo en la fuente – ángulo de elevación (φ) (Cetta, P, 2003).

Figura 3

Planos relativos al oyente y ángulos de posicionamiento de la fuente

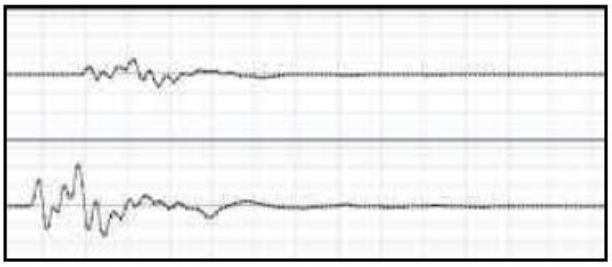


Fuente: (Cetta, P. 2003).

En la figura 4 se observa un gráfico de dos respuestas a impulso, una por cada oído. Estas fueron grabadas con una cabeza artificial. La fuente se ubicó a 30° de azimut y 0° de elevación. El eje horizontal corresponde al tiempo y el vertical a la amplitud de la onda. Se aprecian claramente las diferencias entre ambas. (Cetta, P, 2003).

Figura 4

Respuestas a impulso registradas con una cabeza artificial para $\theta = 30°$ y $\varphi = 0°$

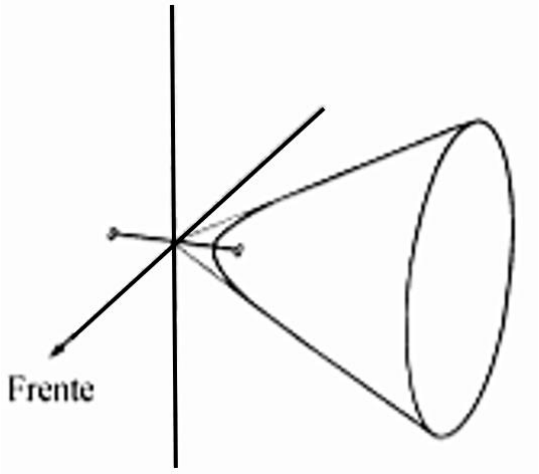


Fuente: (Cetta, P. 2003).

Tanto la ITD como la ILD contribuyen conjuntamente a la lateralización del evento auditivo. Las evaluaciones interauriculares, sin embargo, poseen en sí mismas algunas limitaciones. Cuando la fuente se ubica en el plano medio las diferencias son nulas, y en otros casos la diferencia intramural es la misma para variar posiciones. La figura 5 muestra el denominado “cono de confusión”, en el cual la resta de las distancias entre cualquier punto ubicado en su superficie, y los puntos que representan a los oídos, es constante. (Cetta, P. 2003).

Figura 5

Cono de confusión



(Cetta, P. 2003).

CORRECCIONES ASOCIADAS A LOS MOVIMIENTOS DE LA CABEZA

Las teorías motoras han estudiado en detalle las relaciones entre la posición del evento auditivo y los cambios de las señales de entrada a los oídos durante los movimientos de la cabeza. Estos movimientos, producidos en el momento de la detección de la posición de la fuente, constituyen un indicio adicional de importancia. (Cetta, P. 2003).

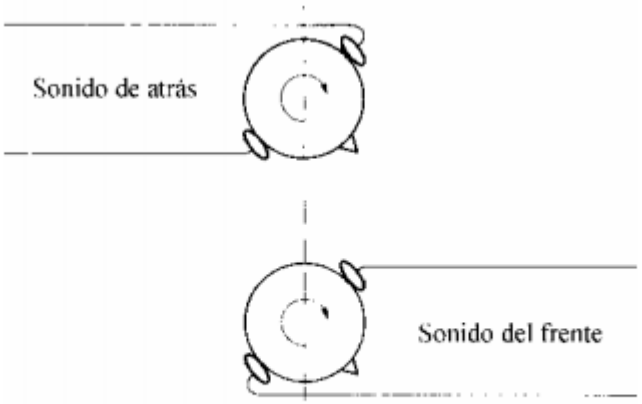
Cuando un oyente mueve su cabeza en la dirección del evento auditivo, buscando la posición probable del evento sonoro, decrece el nivel de indeterminación dado que el mayor grado de precisión se encuentra en la zona frontal. La resolución de las imprecisiones por falta de indicios concretos, o conflicto entre ellos, se realiza a través de los movimientos de la cabeza. (Cetta, P. 2003).

La fuente se ubica sobre el plano medio las diferencias interauriculares son prácticamente nulas. En este caso, el movimiento permite generar una diferencia que sirve, en principio, para

definir si el movimiento proviene del frente o de atrás. La figura 6 puede aclarar este punto. Cuando giramos la cabeza hacia la derecha, si la señal arriba antes al oído derecho, determinamos que la fuente se encuentra atrás. (Cetta, P. 2003).

Figura 6

Indicio aportado por la rotación de la cabeza



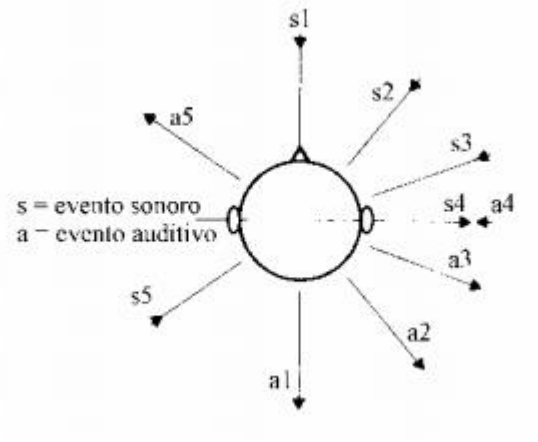
Se atribuye al sistema perceptual la capacidad de reconocer la polaridad de la ITD como elemento aclaratorio de la posición de la fuente. Para este tipo de evaluación es necesario tener presente el sentido de la rotación realizada, que podría estar indicado visualmente, a través del órgano vestibular del equilibrio, o por el estado de los músculos del cuello.

Una anomalía en la detección de la posición de la fuente sonora, especialmente con bandas estrechas, se manifiesta en la percepción del evento en una dirección simétrica respecto al eje que cruza a ambos oídos. Una fuente ubicada en un ángulo de incidencia de 30° sobre el plano horizontal puede ser percibida a 150° (ver figura 7). Como vimos antes, el sistema auditivo recurre a las características espectrales para resolver conflictos entre ambas direcciones, pero cuando los sonidos son de banda angosta, esta información es deficiente o nula, dando lugar a

una falla en la localización, Una manera natural de resolver esta ambigüedad es mediante el movimiento de la cabeza durante la emisión del sonido. En los casos donde la localización es imprecisa el movimiento exploratorio de la cabeza ayuda a resolver el problema.

Figura 7

Percepción simétrica respecto al eje Interaural



2.3.AURILIZACIÓN Y SÍNTESIS BIAURICULAR

La aurilización es el proceso mediante el cual un sonido determinado adopta las características acústicas de un recinto. La aurilización puede llevarse a cabo utilizando modelado acústico a escala, o bien utilizando modelado computarizado. El material fuente, bien sea discurso o música o cualquier sonido, son filtrados por dichas funciones de transferencia por medio de procesamiento digital de señales. La aurilización da la oportunidad de simular la experiencia acústica en cualquier punto de una sala. Es una herramienta fundamental para evaluar la calidad del ambiente acústico de la sala antes de ser construida. (Rey, D., & Gil, L., 2013).

La síntesis biauricular se usa para conectar una señal de sonido a una señal espacial. Para lograr la inmersión del usuario en los sistemas de realidad virtual, es necesario crear sonidos espaciales que coincidan con la impresión espacial visual y otras dimensiones multimodales. (Vorländer, M., & Summers, J. E., 2008)

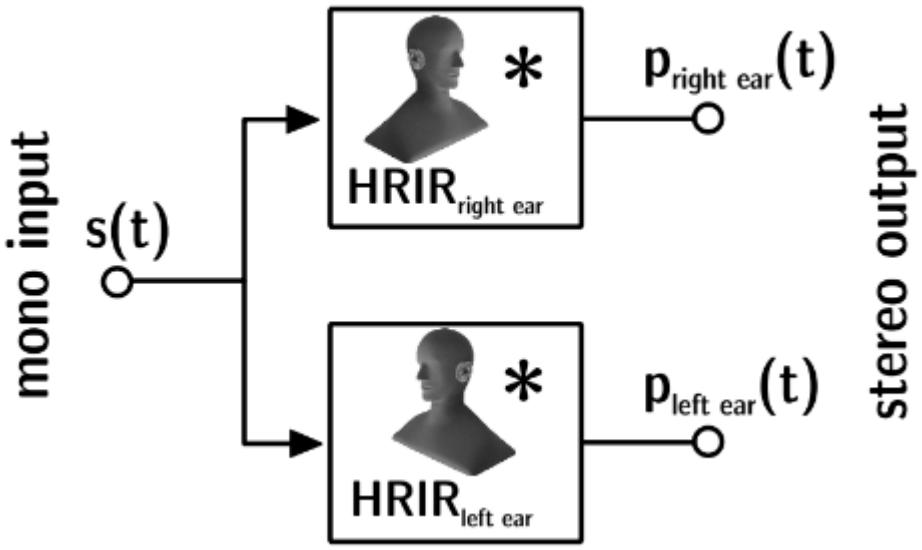
La localización del sonido y la audición espacial pueden entenderse como un efecto de la función de transferencia del oído externo, la función de transferencia relacionada con la cabeza (HRTF). Con una base de datos de la HRTF, se puede simular cualquier dirección de incidencia del sonido, cuando una fuente mono s(t) se convoluciona con un par de respuestas de impulso relacionadas con él. (Vorländer, M., & Summers, J. E., 2008)

$$p_{\text{right ear}}(t) = s(t) * HRTF_{\text{right ear}}$$
$$p_{\text{left ear}}(t) = s(t) * HRTF_{\text{left ear}}$$

La tarea básica para crear una auralización es colocar una fuente de sonido en un espacio 3D cualquiera señal de fuente mono debidamente caracterizada y calibrada, puede procesarse de modo que sus señales de percepción sean modificadas por un componente espacial. Una configuración estéreo o envolvente es capaz de crear un efecto de fuentes que puede producir un efecto espacial apropiado. Se puede usar una consola de mezclas biauricular para procesar señales de auriculares usando HRTF. (Vorländer, M., & Summers, J. E., 2008)

Figura 8

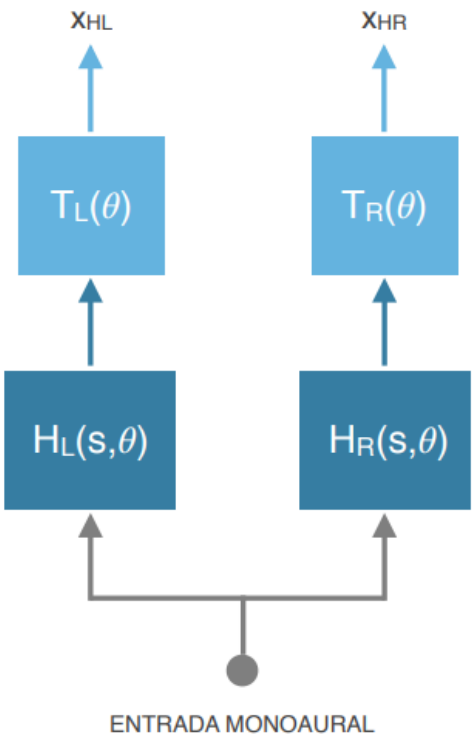
Síntesis biauricular



Fuente: Vorländer, M., & Summers, J. E. (2008).

Figura 9

Esquema de aplicación del Modelo de cabeza



Nota. La figura representa el Esquema de aplicación del Modelo de cabeza. HL (s, θ) y HR (s, θ) son el resultado de aplicación de la función de transferencia para cada uno de los canales. TL (θ) y TR (θ) son los retrasos correspondientes al ITD. Fuente: (Recodo Estévez, S. 2017).

2.4. APRENDIZAJE AUTOMÁTICO SUPERVISADO Y NO SUPERVISADO.

El objetivo del aprendizaje automático es mejorar las habilidades de un agente. El aprendizaje entra en juego cuando el agente inteligente observa sus interacciones con el mundo y en base a estas hace una toma de decisiones. (Russell, S., & Norvig, P, 2004)

Hay diferentes formas en las que nosotros mismos aprendemos de nuestros entornos circundantes. En un sentido amplio, se puede categorizar los procesos de aprendizaje a través de los cuales funciona el Machine Learning: aprender con un maestro y aprender sin un maestro. Estas formas de aprendizaje son paralelas a las del aprendizaje humano. (Haykin, 2008)

APRENDIZAJE SUPERVISADO

El objetivo de este aprendizaje es crear un modelo a partir de inputs y sus correspondientes outputs. Para que el modelo aprenda se utiliza características y una etiqueta que suministra el valor correcto de la salida de cada entrada. El valor de la salida proviene de lo que el agente percibe. En entornos totalmente observables, el agente siempre puede observar los efectos de sus acciones, y, por lo tanto, puede utilizar métodos de aprendizaje supervisado para aprender a predecirlos. En entornos que son parcialmente observables, el problema es más difícil, ya que los efectos más inmediatos pueden ser invisibles. (Russell & Norvig, 2004)

La Figura 10 muestra un diagrama de bloques que ilustra esta forma de aprendizaje. En términos conceptuales, se puede pensar que hay un maestro que tiene conocimiento del entorno, con ese conocimiento representado por un conjunto de ejemplos de entrada-salida lo que se puede denominar como un dataset. Los parámetros del modelo se ajustan bajo la influencia combinada del vector de entrenamiento y la señal de error. La señal de error se define como la diferencia entre la respuesta deseada y la respuesta real del modelo. Este ajuste se lleva a cabo de forma iterativa, paso a paso, con el objetivo de hacer eventualmente que el modelo emule al profesor. De esta manera, el conocimiento del entorno disponible para el profesor se transfiere al modelo a través del entrenamiento y se almacena en forma de pesos sinápticos "fijos", que representan como la memoria a largo plazo. Cuando se alcanza esta condición, se puede

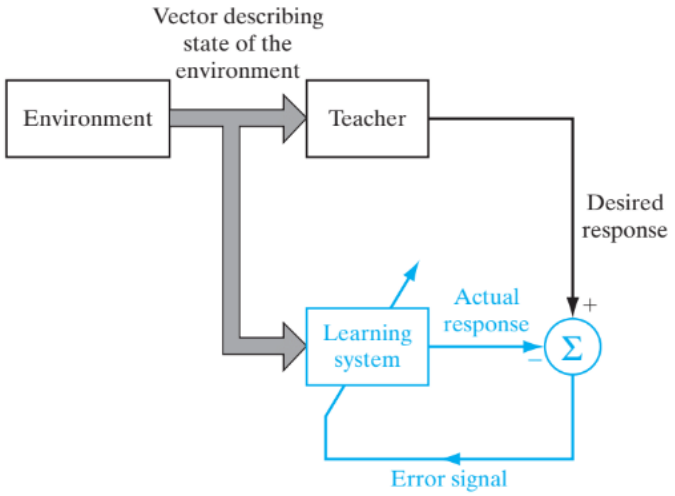
prescindir del maestro y dejar que el modelo se ocupe del entorno completamente por sí mismo. (Haykin 2008)

La forma de aprendizaje supervisado anteriormente descrito es la base de aprendizaje de corrección de errores. (Haykin 2008)

En la Figura 10, se puede ver que el proceso de aprendizaje supervisado constituye un sistema de retroalimentación de circuito cerrado. Como medida de desempeño del sistema, se puede pensar en términos del error cuadrático medio o en la suma de los errores cuadráticos sobre la muestra de entrenamiento. Esta función puede visualizarse como una superficie de comportamiento de error multidimensional, o simplemente una superficie de error, con los parámetros libres como coordenadas. La verdadera superficie de error se promedia sobre todos los posibles ejemplos de entrada y salida. Cualquier operación dada del sistema bajo la supervisión del maestro se representa como un punto en la superficie del error. Para que el sistema mejore el rendimiento con el tiempo y, por lo tanto, aprenda del profesor, el punto de operación tiene que descender sucesivamente hacia un punto mínimo de la superficie de error; el punto mínimo puede ser un mínimo local o un mínimo global. Un sistema de aprendizaje supervisado es capaz de hacer esto con la información útil que tiene sobre el gradiente de la superficie de error correspondiente al comportamiento actual del sistema. El gradiente de la superficie de error en cualquier punto es un vector que apunta en la dirección de mayor pendiente, descendencia. (Haykin 2008)

Figura 10

Esquema de aplicación del Modelo de cabeza



Fuente: Haykin (2008).

APRENDIZAJE NO SUPERVISADO

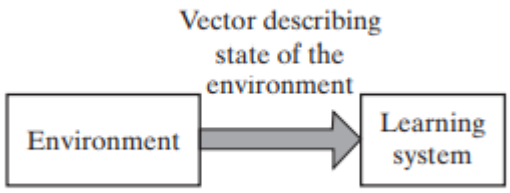
En este tipo de aprendizaje se obtiene patrones de entradas para los que no se especifican los valores de sus salidas. Como ejemplo un agente taxista debería desarrollar gradualmente los conceptos de «días de tráfico bueno» y de «días de tráfico malo», sin que le hayan sido dados ejemplos etiquetados de ello. (Russell & Norvig, 2004).

En el aprendizaje supervisado, el proceso de aprendizaje se lleva a cabo bajo la tutela de un maestro. Sin embargo, en el paradigma conocido como aprendizaje sin maestro, como su nombre lo indica, no hay un maestro que supervise el proceso de aprendizaje, es decir, no hay ejemplos etiquetados de la función que debe aprender el agente. En el dataset que se tiene simplemente existirán entradas sin ninguna etiqueta o salida que defina el valor de las entradas. (Haykin 2008)

En el aprendizaje no supervisado o auto organizado, no hay un profesor o crítico externo que supervise el proceso de aprendizaje, como se indica en la figura 11. Más bien, se prevé una medida independiente de la tarea de la calidad de representación del modelo. Requerido para aprender, y los parámetros libres de la red se optimizan con respecto a esa medida. Para una medida específica independiente de la tarea, una vez que el modelo se ha sintonizado con las regularidades estadísticas de los datos de entrada, la red desarrolla la capacidad de formar representaciones internas para codificar características de la entrada y, por lo tanto, crear nuevas clases automáticamente. (Haykin 2008)

Figura 11

Diagrama de bloques de aprendizaje no supervisado



Fuente: Haykin (2008).

2.5. TÉCNICAS CLASIFICACIÓN MACHINE LEARNING

Una vez generado el modelo, la salida de este será una clase. Esta clase será entre un número limitado de zonas de procedencia de sonido. Al hablar de clase se refiere a categorías arbitrarias. Para el caso de este proyecto cada categoría será una zona determinada alrededor de la cabeza que es determinada por el ángulo de azimut y el ángulo de elevación.

Entre las diferentes técnicas y algoritmos de clasificación para machine learning encontramos varias opciones. Sin embargo, para la elaboración de este proyecto se considerarán los siguientes algoritmos:

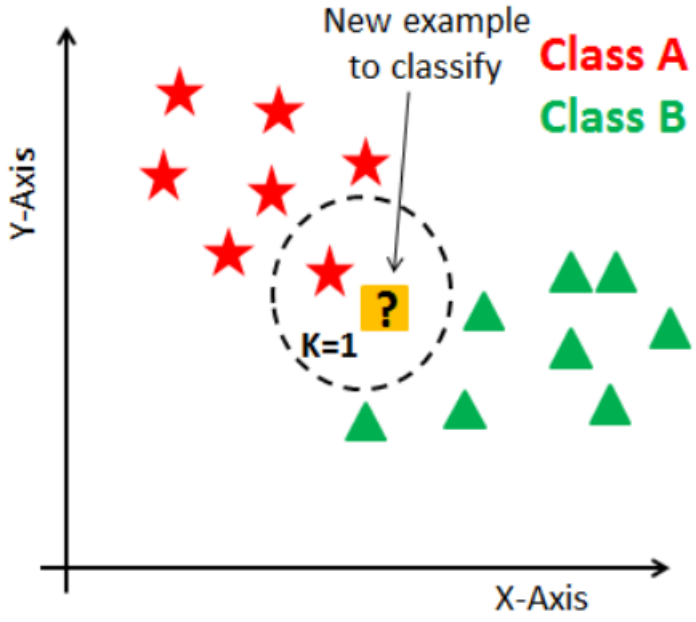
KN- Neighbors: “La clasificación de vecinos más cercanos, también conocida como K-nearest neighbors más cercanos (KNN), es basado los patrones más cercanos a un patrón objetivo x, por lo cual se busca la etiqueta, se entrega información útil de la etiqueta. KNN asigna la clase etiqueta de la mayoría de los K-patrones más cercanos en el espacio de datos. Por este motivo, se debe ser capaz de definir una medida de similitud en el espacio de datos.” (Kramer, 2013)

La manera cómo funciona KNN es que calcula la distancia que existe entre la muestra para clasificar y los demás registros del dataset para entrenamiento. A partir de aquí selecciona los “k” elementos más cercanos con menos distancia. La forma en cómo se calcula la distancia depende de la función que se use (Manhattan, Euclidiana, etc.). Finalmente, se revisa cual es la clase o etiqueta que más predomina entre los elementos que se ha seleccionado para categorizar la muestra.

Este algoritmo se caracteriza por ser aprendizaje supervisado y que se basa en instancias, es decir que no aprende explícitamente un modelo para poder categorizar el input. En cambio, este algoritmo usa una base de conocimiento que es el dataset usado para el entrenamiento y a partir de aquí realiza la fase de predicción.

Figura 12

Clasificación de vecinos más cercanos



Fuente: Navlani (2018).

Tree: "Un árbol de decisión es una forma gráfica y analítica de representar todos los eventos (sucesos) que pueden surgir a partir de una decisión asumida en cierto momento. Permiten tomar la decisión más "acertada", desde un punto de vista probabilístico, ante un abanico de posibles decisiones. Estos árboles permiten examinar los resultados y determinar visualmente cómo fluye el modelo. Los resultados visuales ayudan a buscar subgrupos específicos y relaciones que tal vez son más difíciles de encontrar con estadísticos más tradicionales." (Berlanga, Rubio, Vilà, 2013)

Los árboles de decisión son una técnica estadística para la segmentación, la estratificación, la predicción, la reducción de datos y el filtrado de variables, la identificación de interacciones, la fusión de categorías y la discretización de variables continuas. (Berlanga, Rubio, Vilà, 2013)

Los árboles de decisión crean un modelo de clasificación basado en diagramas de flujo. Clasifican casos en grupos o pronostican valores de una variable dependiente (criterio) basada en valores de variables independientes (predictores). (Berlanga, Rubio, Vilà 2013).

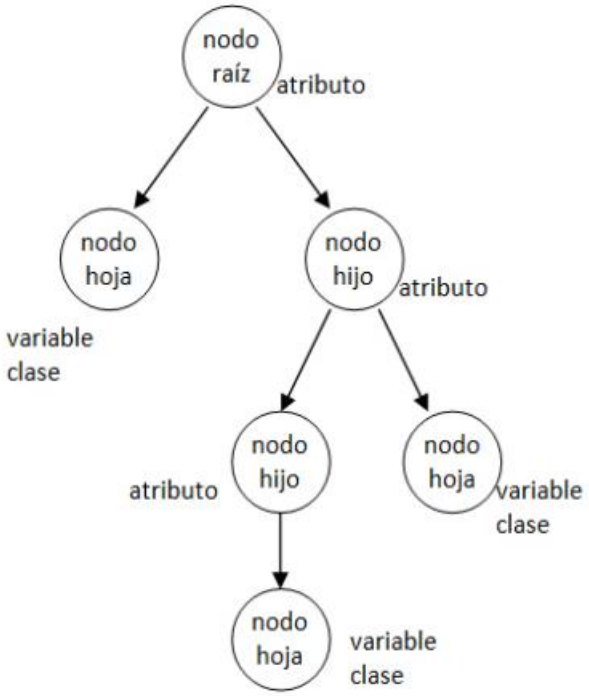
“Un árbol de decisión es un modelo de predicción cuyo objetivo principal es el aprendizaje inductivo a partir de observaciones y construcciones lógicas. Son muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que suceden de forma sucesiva para la solución de un problema. Constituyen probablemente el modelo de clasificación más utilizado y popular. El conocimiento obtenido durante el proceso de aprendizaje inductivo se representa mediante un árbol. Un árbol gráficamente se representa por un conjunto de nodos, hojas y ramas. El nodo principal o raíz es el atributo a partir del cual se inicia el proceso de clasificación; los nodos internos corresponden a cada una de las preguntas acerca del atributo en particular del problema. Cada posible respuesta a los cuestionamientos se representa mediante un nodo hijo. Las ramas que salen de cada uno de estos nodos se encuentran etiquetadas con los posibles valores del atributo 2. Los nodos finales o nodos hoja corresponden a una decisión, la cual coincide con una de las variables clase del problema a resolver (Ver Figura 13)”. (Martínez R, y otros, 2009)

“Este modelo se construye a partir de la descripción narrativa de un problema, ya que provee una visión gráfica de la toma de decisión, especificando las variables que son evaluadas, las acciones que deben ser tomadas y el orden en el que la toma de decisión será efectuada. Cada

vez que se ejecuta este tipo de modelo, sólo un camino será seguido dependiendo del valor actual de la variable evaluada. Los valores que pueden tomar las variables para este tipo de modelos pueden ser discretos o continuos.” (Martínez, Ramírez, Mesa, Suárez, Trejo, León, Morales, 2009).

Figura 13

Estructura de un árbol de decisión



Fuente: Martínez, Ramírez, Mesa, Suárez, Trejo, León, Morales (2009).

“Un algoritmo de generación de árboles de decisión consta de 2 etapas: la primera corresponde a la inducción del árbol y la segunda a la clasificación. En la primera etapa se construye el árbol de decisión a partir del conjunto de entrenamiento; comúnmente cada nodo interno del árbol se compone de un atributo de prueba y la porción del conjunto de entrenamiento

presente en el nodo es dividida de acuerdo con los valores que pueda tomar ese atributo." (Martínez, Ramírez, Mesa, Suárez, Trejo, León, Morales 2009).

"La construcción del árbol inicia generando su nodo raíz, eligiendo un atributo de prueba y dividiendo el conjunto de entrenamiento en dos o más subconjuntos; para cada partición se genera un nuevo nodo y así sucesivamente. Cuando en un nodo se tienen objetos de más de una clase se genera un nodo interno; cuando contiene objetos de una clase solamente, se forma una hoja a la que se le asigna la etiqueta de la clase. En la segunda etapa del algoritmo cada objeto nuevo es clasificado por el árbol construido; después se recorre el árbol desde el nodo raíz hasta una hoja, a partir de la que se determina la membresía del objeto a alguna clase. El camino para seguir en el árbol lo determinan las decisiones tomadas en cada nodo interno, de acuerdo con el atributo de prueba presente en él." (Martínez, Ramírez, Mesa, Suárez, Trejo, León, Morales 2009).

Random Forest: Un modelo de Random Forest es un conjunto de árboles de decisión individuales. Los cuales se obtienen tras un proceso de entrenamiento en el que se utiliza una muestra aleatoria diferente para cada uno de ellos. Obteniendo así una predicción diferente en cada uno de los árboles.

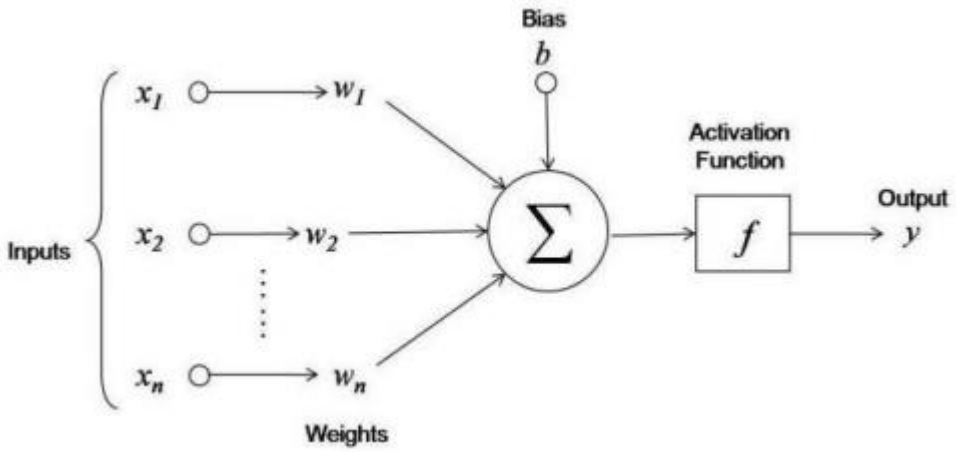
Algunas de las ventajas de utilizar Random Forest es que reduce la varianza esto "debido a la utilización de múltiples árboles de decisión la varianza de la predicción disminuye respecto a un solo árbol de decisión. Lo que reduce la posibilidad de obtener un modelo sobre ajustado. Aun así, el sesgo del modelo es el mismo que se obtiene con un solo árbol de decisión. No requiere la asunción de suposiciones. Cuando solamente se quiere un modelo para realizar predicciones, Random Forest es una solución excelente. Esto es así porque no es necesario

realizar suposiciones sobre el modelo o los conjuntos de datos.” ("Random Forest - Analytics Lane", 2022)

Neural Network: Una red neuronal es un conjunto de unidades, mismas que se denominan neuronas artificiales. Estas neuronas se interconectan entre sí para poder transmitir señales. Existen datos de entrada los cuales se someten a unas operaciones y finalmente la neurona genera datos de salida.

Figura 14

Modelo genérico de una neurona artificial



Fuente: Artola Moreno (2019)

En el modelo de neurona artificial podemos observar:

Sinapsis: La sinapsis en la neurona se representa por la conexión o el conjunto de conexiones (inputs), mismos que tienen un peso representado por la letra ‘W’. Estos pesos sinápticos de la neurona podrían tener desde valores negativos como valores positivos.

La función de red: Esta función se encarga de calcular la suma de las señales de entrada, estas ponderadas por las respectivas sinapsis de la neurona.

La función de activación: esta función permite transformar el valor de la función de red generar un nuevo valor de salida. Esta "limita el rango de amplitud permitido en la salida a un valor finito." (Artola Moreno, 2019)

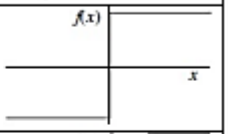
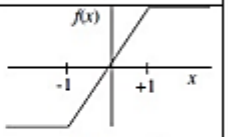
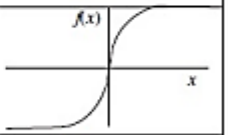
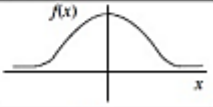
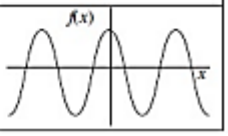
La salida: Esta es el resultado de la función de activación.

Funciones de activación:

En las funciones de activación se tiene las siguientes:

Figura 15

Funciones de activación

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty,+\infty]$	
Escalón	$y = sign(x)$ $y = H(x)$	$\{-1,+1\}$ $\{0,+1\}$	
Lineal a tramos	$y = \begin{cases} -1, si\ x < -1 \\ x, si\ +1 \le x \le -1 \\ +1, si\ x > +1 \end{cases}$	$[-1,+1]$	
Sigmoidea	$y = \frac{1}{1+e^{-x}}$ $y = tgh(x)$	$[0,+1]$ $[-1,+1]$	
Gaussiana	$y = Ae^{Bx^2}$	$[0,+1]$	
Sinusidal	$y = A\,sen(\omega x + \varphi)$	$[-1,+1]$	

Fuente: Artola Moreno (2019)

Las redes neuronales están compuestas por una o varias capas, a partir de esto se puede clasificar las redes neuronales en:

Figura 16

Red neuronal monocapa

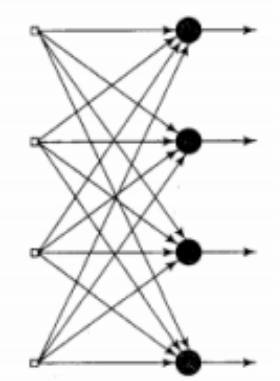
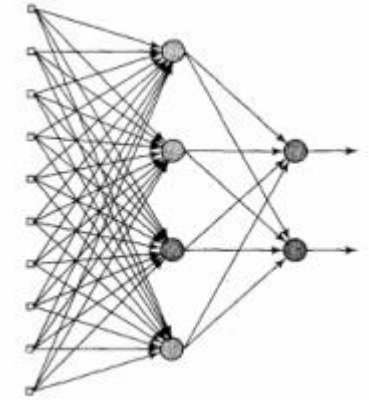


Figura 17

Red Neuronal multicapa



Fuente: Artola Moreno (2019)

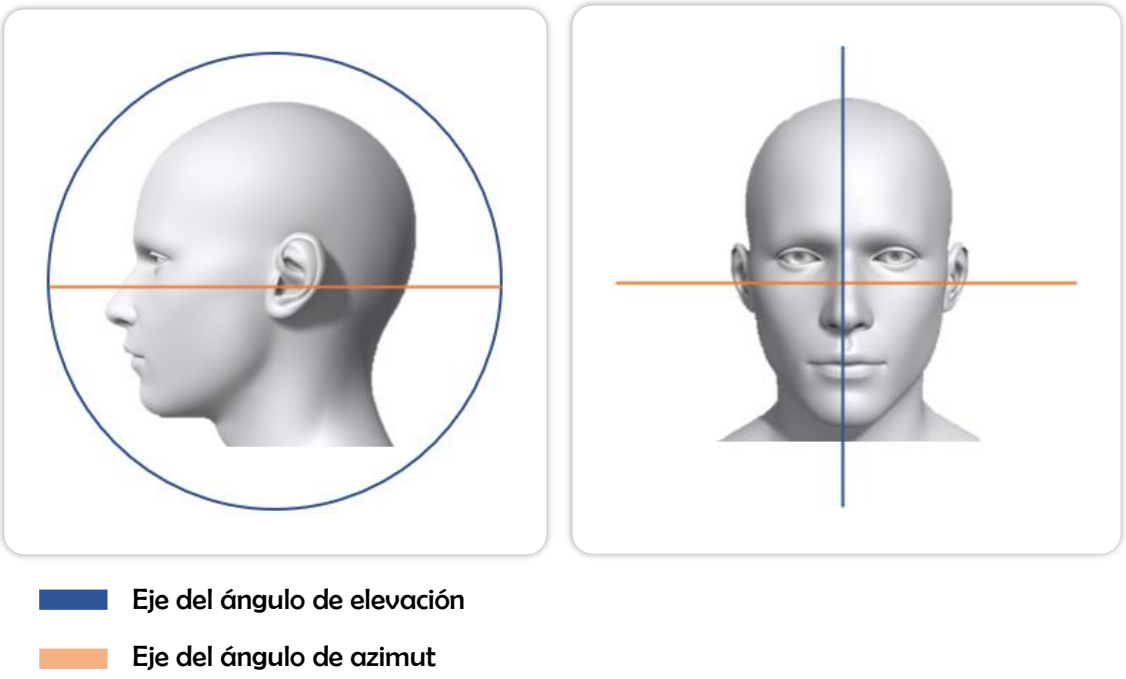
3. METODOLOGÍA Y MATERIALES

3.1. DEFINICIÓN DEL DATASET DE SEÑALES DIRECCIONALES

Para este proyecto se requiere crear un dataset de señales direccionales de diferentes ángulos de elevación como de ángulos de azimut. Este dataset sirve poder entrenar el modelo. Dado que para poder conocer la procedencia direccional de un sonido se necesita hacer uso de los ángulos, se ocupa una esfera la cual representa todos los posibles lugares de los cuales un sonido pueda provenir Figura 18.

Figura 18

Representación de la esfera formada por los ejes de los ángulos de elevación y azimut



Fuente: Realización propia

Como se puede observar la esfera se encuentra formada por los ejes de los dos ángulos que definen la posición de una fuente sonora. Para la generación de este dataset se ha tomado una base de HRTF's (2.2) resultado de un trabajo elaborado por el MIT en donde se tomó las muestras de sonidos de varias direcciones con micrófonos simulando la posición de una cabeza

humana (KEMAR). Las emisiones de sonido fueren montadas a 1,4 metros del centro del KEMAR. Se utilizaron secuencias binarias pseudoaleatorias de longitud máxima (ML) para obtener la respuesta de impulso a una frecuencia de muestreo de 44,1 kHz. En total, se muestrearon 710 posiciones diferentes en elevaciones entre -40 grados a 90 grados (Figura 19). También se midió la respuesta al impulso del altavoz en campo libre y varios auriculares colocados en el KEMAR. Estos datos están disponibles para la comunidad de investigación en Internet a través de FTP anónimo y el World Wide Web. (Gardner, B. y Martin, K. 1994) La data puede ser obtenida del siguiente enlace: https://sound.media.mit.edu/resources/KEMAR.html

Figura 19

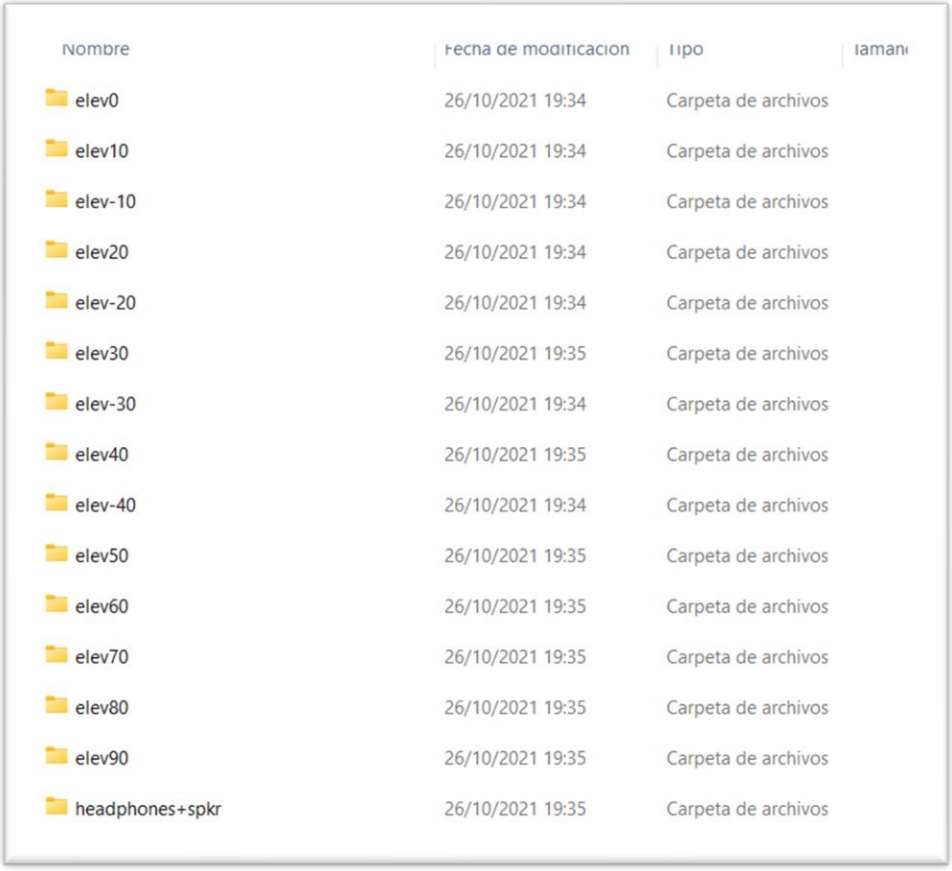
Muestras de los diferentes ángulos de elevación

Elevation	Number of Measurements	Azimuth Increment
-40	56	6.43
-30	60	6.00
-20	72	5.00
-10	72	5.00
0	72	5.00
10	72	5.00
20	72	5.00
30	60	6.00
40	56	6.43
50	45	8.00
60	36	10.00
70	24	15.00
80	12	30.00
90	1	x

Nota. Muestras obtenidas de los diferentes ángulos de elevación como de Azimut Tomado de Fuente: Gardner, B. y Martin, K. (1994).

Figura 20

Muestras de los diferentes ángulos de elevación

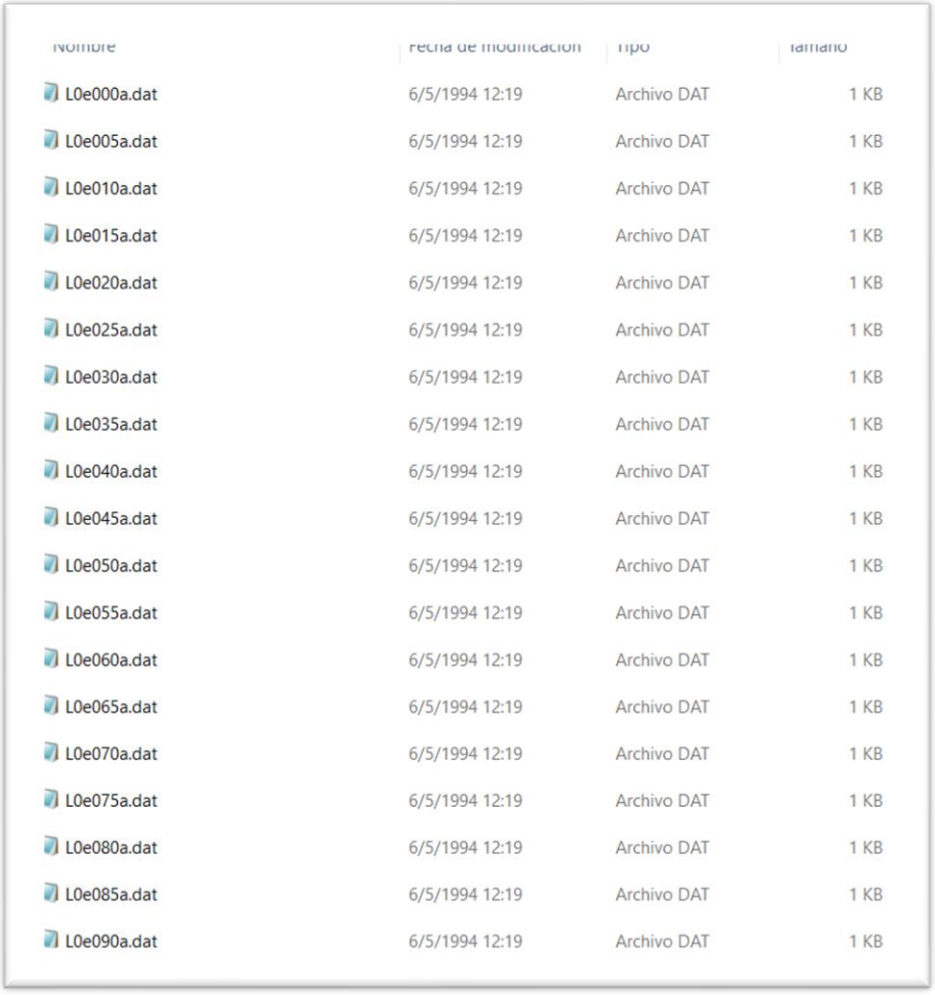


Nombre	Fecha de modificación	Tipo	Tamaño
elev0	26/10/2021 19:34	Carpeta de archivos	
elev10	26/10/2021 19:34	Carpeta de archivos	
elev-10	26/10/2021 19:34	Carpeta de archivos	
elev20	26/10/2021 19:34	Carpeta de archivos	
elev-20	26/10/2021 19:34	Carpeta de archivos	
elev30	26/10/2021 19:35	Carpeta de archivos	
elev-30	26/10/2021 19:34	Carpeta de archivos	
elev40	26/10/2021 19:35	Carpeta de archivos	
elev-40	26/10/2021 19:34	Carpeta de archivos	
elev50	26/10/2021 19:35	Carpeta de archivos	
elev60	26/10/2021 19:35	Carpeta de archivos	
elev70	26/10/2021 19:35	Carpeta de archivos	
elev80	26/10/2021 19:35	Carpeta de archivos	
elev90	26/10/2021 19:35	Carpeta de archivos	
headphones+spkr	26/10/2021 19:35	Carpeta de archivos	

Nota. Contenido de la data descargada dividido en carpetas según el ángulo de Elevación. Fuente: Realización propia (2021)

Figura 21

Contenido de la carpeta "elev0"



Nombre	Fecha de modificación	Tipo	Tamaño
L0e000a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e005a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e010a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e015a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e020a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e025a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e030a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e035a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e040a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e045a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e050a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e055a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e060a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e065a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e070a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e075a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e080a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e085a.dat	6/5/1994 12:19	Archivo DAT	1 KB
L0e090a.dat	6/5/1994 12:19	Archivo DAT	1 KB

Nota. Contenido de la carpeta "elev0" en donde se encuentran los datos de las muestras tomadas desde un ángulo de elevación con valor cero y diferentes ángulos de azimut. Fuente: Realización propia (2021).

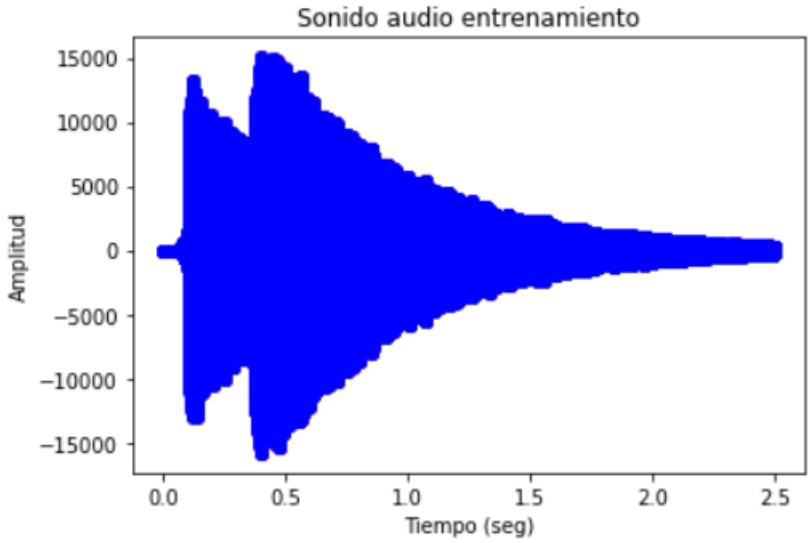
Una vez obtenidas las muestras de diferentes ángulos que conforman la esfera antes mostrada, debemos generar el dataset de sonidos procedentes de diferentes direcciones.

El dataset de señales direccionales que se creará es uno conjunto de audios con extensión ‘.wav’ en los que cada audio se reproduce con un ángulo de elevación y un ángulo de azimut diferente.

Para la creación de los audios con sus respectivas direcciones se debe partir de un audio, mismo que será de un sonido repentino como lo es el sonido de un timbre de una casa. Este sonido será de alrededor de dos segundos y medio. El audio que se utilizará tendrá un sonido mono auricular, es decir el sonido y la señal será la misma por el lado derecho e Izquierdo. Este audio se utilizará para genera varios audios biauriculares que tengan las características propias de procedencia de diferentes direcciones.

Figura 22

Gráfico de la onda del sonido de un timbre



Fuente: Realización propia (2021).

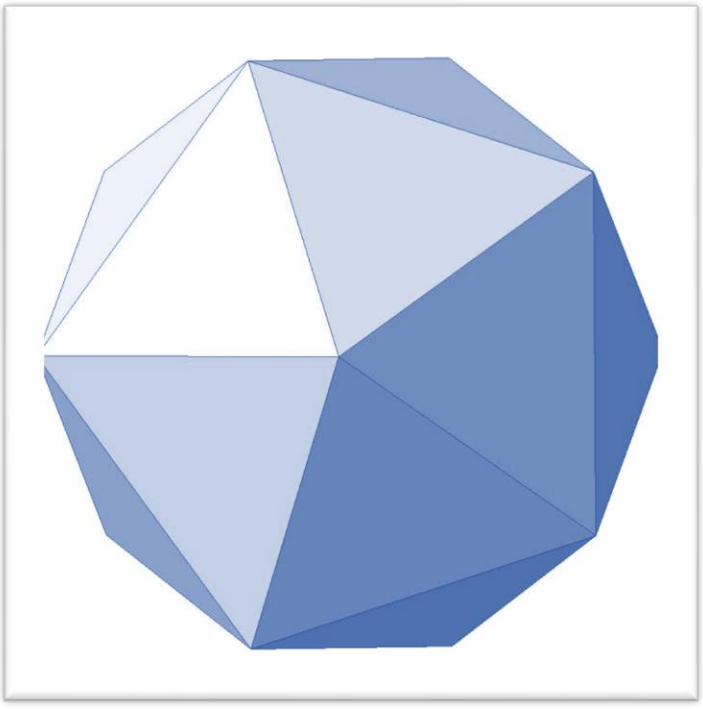
Para la creación del dataset se utilizará el software Matlab en donde gracias a su entorno de trabajo de escritorio permite procesar audio y funciones matemáticas para los cálculos.

En primer lugar, es necesario definir las direcciones que tendrán los diferentes audios del dataset. Estas direcciones son el ángulo de azimut y el ángulo de elevación. Para esto se toma como referencia la esfera formada por estos ángulos (figura 18). Esta será la base para extraer las direcciones de los diferentes puntos que se pueden extraer del borde de la esfera. Lo ideal es tener un conjunto de audios uniformes, es decir que los puntos de los que se vaya a tomar las direcciones deben estar a la misma distancia uno de otro alrededor de la esfera.

Para resolver el problema de uniformidad antes descrito se tomará como puntos iniciales los vértices de un icosaedro (figura 23).

Figura 23

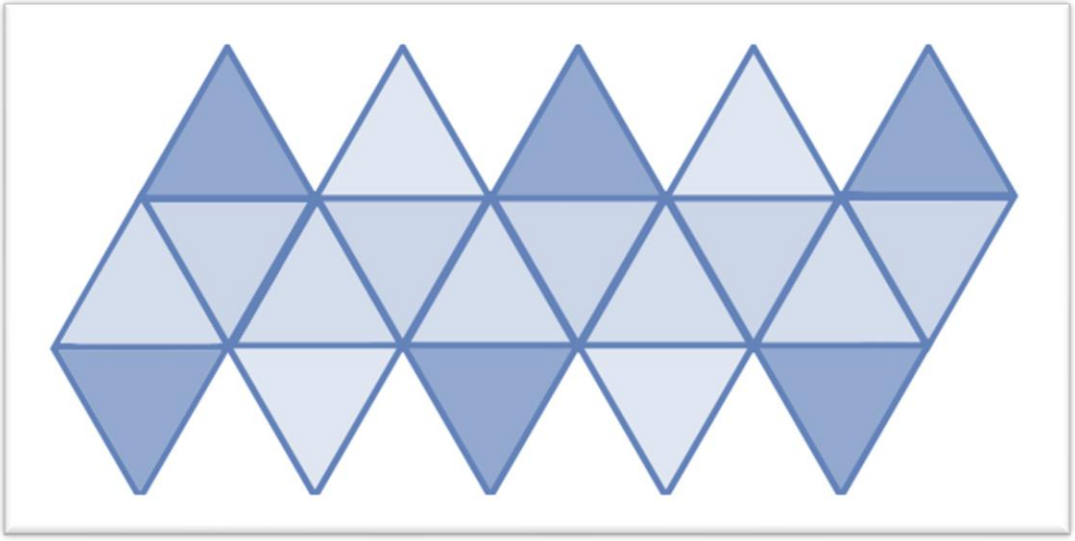
Icosaedro



Fuente: Realización propia (2021).

Figura 24

Icosaedro abierto



Fuente: Realización propia (2021).

Dado que los vértices del icosaedro son uniformes entre sí, estos sirven como base para generar los demás puntos. Para la extracción de los demás puntos se utiliza la función ‘createRays’ desarrollada por el Ingeniero José Lucio (Lucio, 2021) (véase anexo 1). Esta función toma como parámetro el número de puntos que se van a generar y devuelve tres vectores de puntos en coordenadas cartesianas.

Figura 25

Imagen de los vectores resultado de la función createRays ( )

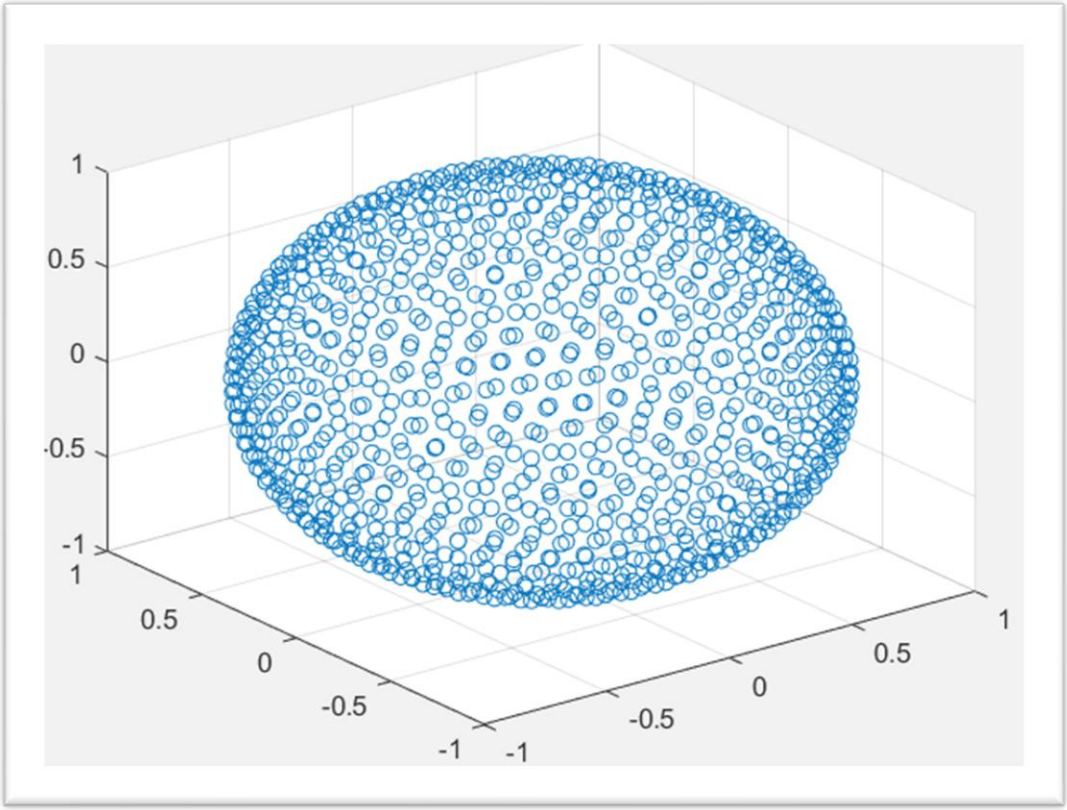


Fuente: Realización propia (2021).

Una vez generado los puntos, se necesita graficarlos y comprobar visualmente la distancia uniforme entre estos. De este modo utilizando Matlab se hace uso de la función *scatter3(vector_x, vector_y, vector_z)* en donde se coloca como parámetros los vectores en coordenadas escalares obtenidas en el anterior procedimiento.

Figura 26

Gráfico de la esfera con radio=1 unidad

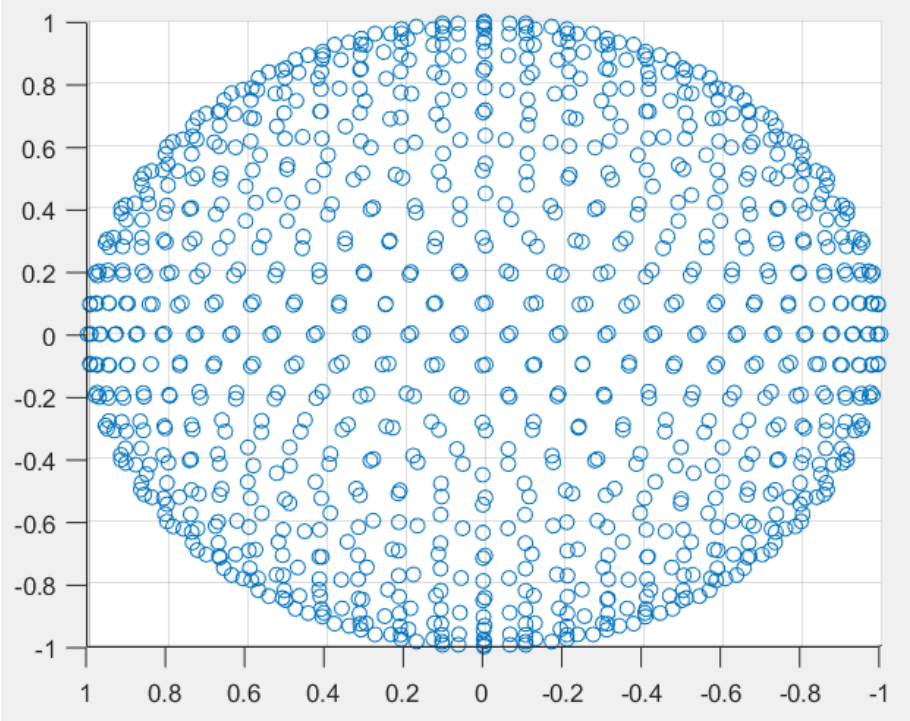


Nota. Gráfico de la esfera con radio=1 unidad, conformada por los puntos generados previamente.

Fuente: Realización propia (2021).

Figura 27

Vista frontal de la esfera



Fuente: Realización propia (2021).

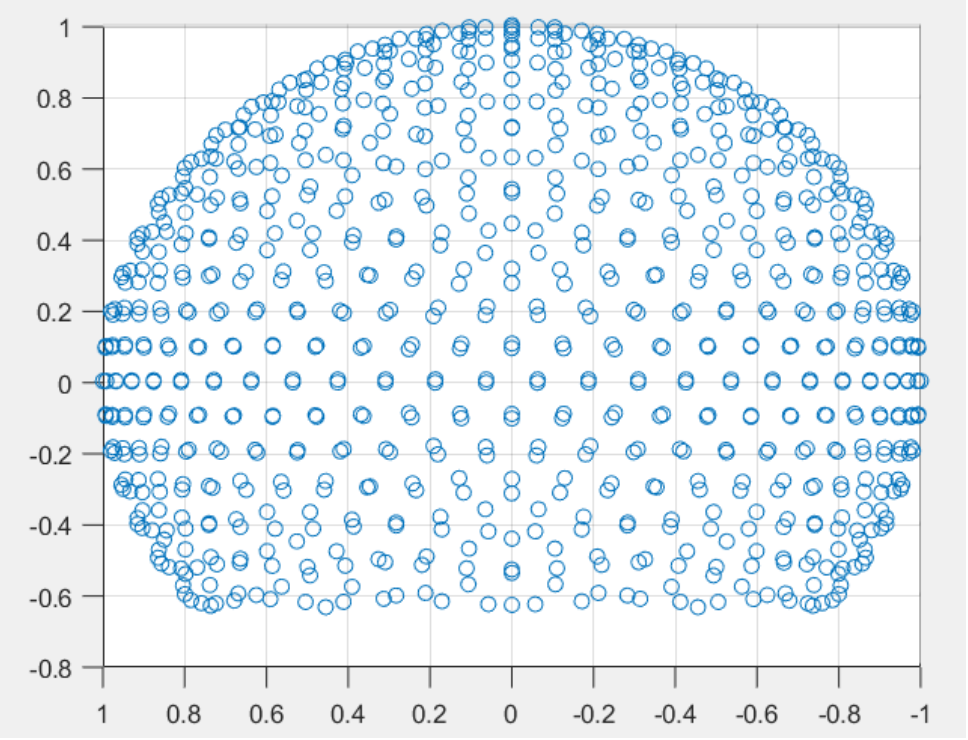
La esfera que se generó tiene los puntos distribuidos uniformemente por todas las direcciones. Estos puntos proporcionan los ángulos para generar los audios. Estos audios son generados a partir de una interpolación entre el audio base para el entrenamiento y de las HRTF's generadas por el MTI. Estas muestras tienen distribuido el ángulo de azimut por los 360 grados, pero el Angulo de elevación va desde 90° hasta los - 40°. Por este motivo, se debe eliminar aquellos puntos de la esfera que tengan una dirección por debajo de los -40° de azimut.

Para la eliminación de los puntos que no se va a necesitar, primero se debe transformar estos puntos de coordenadas cartesianas a coordenadas polares. Esta transformación se realiza mediante la función *cart2sph(vector_x, vector_y, vector_z)* en donde ingresamos como parámetros los vectores de las tres dimensiones de los puntos en coordenadas escalares. El

resultado de esta función es un vector de los ángulos de azimut, un vector de los ángulos de elevación y un vector de los radios. Una vez que se tiene las direcciones de cada punto en coordenadas polares hay que transformarlas de radianes a ángulos. Esto se lo hace simplemente multiplicando los radianes por 180 y dividirlo para PI. De esta manera ya se tiene los ángulos de azimut en grados y se realiza la eliminación de los puntos que no se necesita. De este modo se vuelve a generar la esfera sin los puntos sobrantes teniendo como resultado la siguiente imagen (Figura 28).

Figura 28

Esfera generada por puntos



Nota. Esfera generada por puntos excepto aquellos que tienen como dirección un ángulo de azimut menor a -40° Fuente: Realización propia (2021).

Una vez se tiene las direcciones de las cuales procederán los sonidos, se realiza la creación de los audios. Para esto se debe hacer una interpolación. Se toma el audio mono auricular base (Figura 29) al cual se debe hacer un escalamiento al exponente de la próxima potencia superior de dos para realizar la Transformada de Fourier.

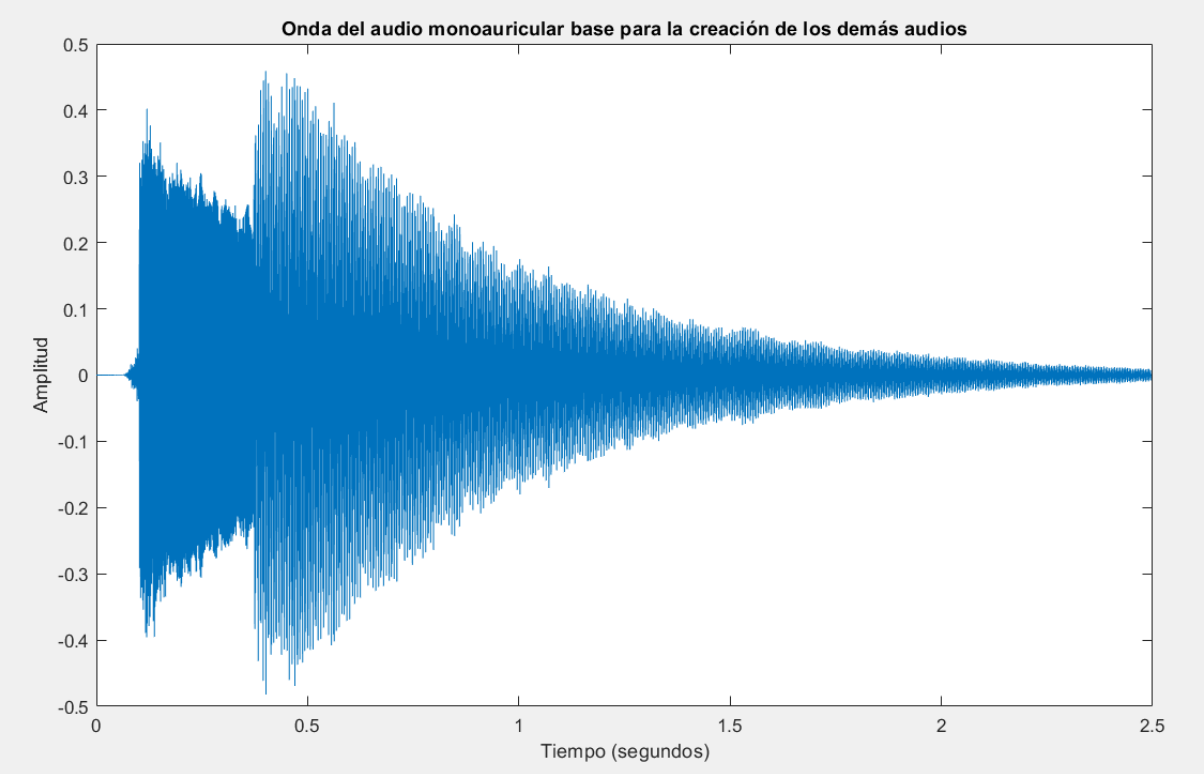
Transformada de Fourier: Y = fft(X) implementa la transformada de Fourier. Y de longitud n, esta transformada se define de la siguiente manera:

$$Y(k) = \sum_{j=1}^{n} X(j)\, W_n^{(j-1)(k-1)}$$

Donde $W_n = e^{(-2\pi i)/n}$ es una de **n** raíces de unidad

Figura 29

Onda del audio mono auricular base



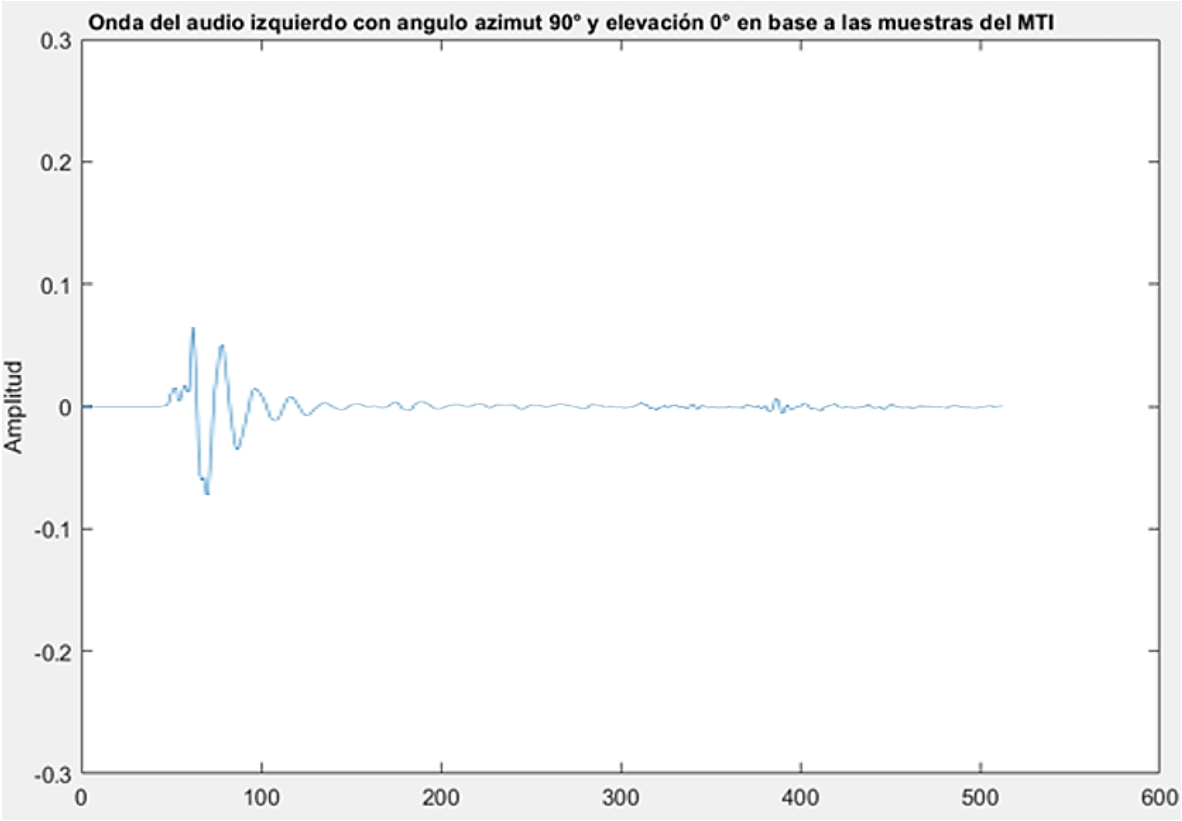
Nota. En esta figura se muestra la onda del audio mono auricular base para la creación de los demás audios.

Fuente: Realización propia (2021).

Una vez que se tiene la trasformada de Fourier del audio mono auricular base se debe extraer la transformada de Fourier de las HRTF's realizadas por el MTI. Este proceso se lo realiza del mismo modo que con el audio, con la diferencia que para este caso se hace esta operación para cada señal de ambos auriculares.

Figura 30

Onda del audio izquierdo

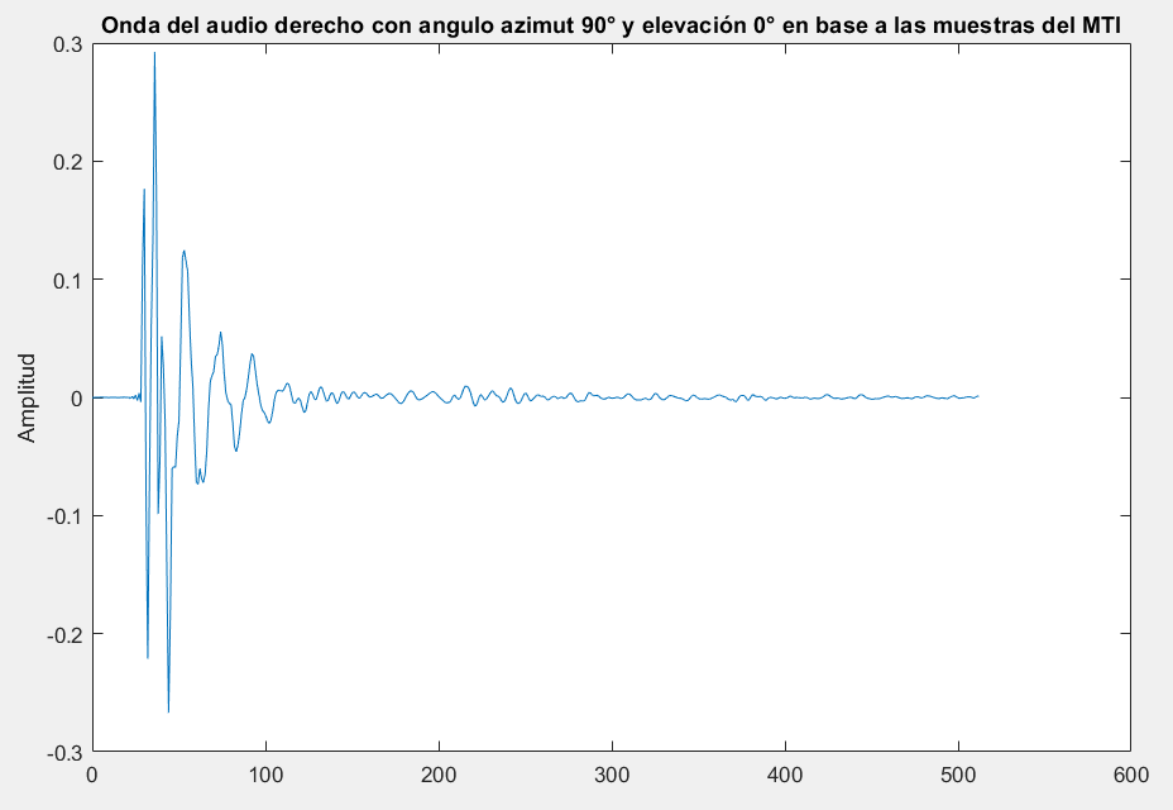


Nota. Onda del audio izquierdo con ángulo azimut 90° y elevación 0° en base a las muestras del MTI

Fuente: Realización propia (2021).

Figura 31

Onda del audio derecho



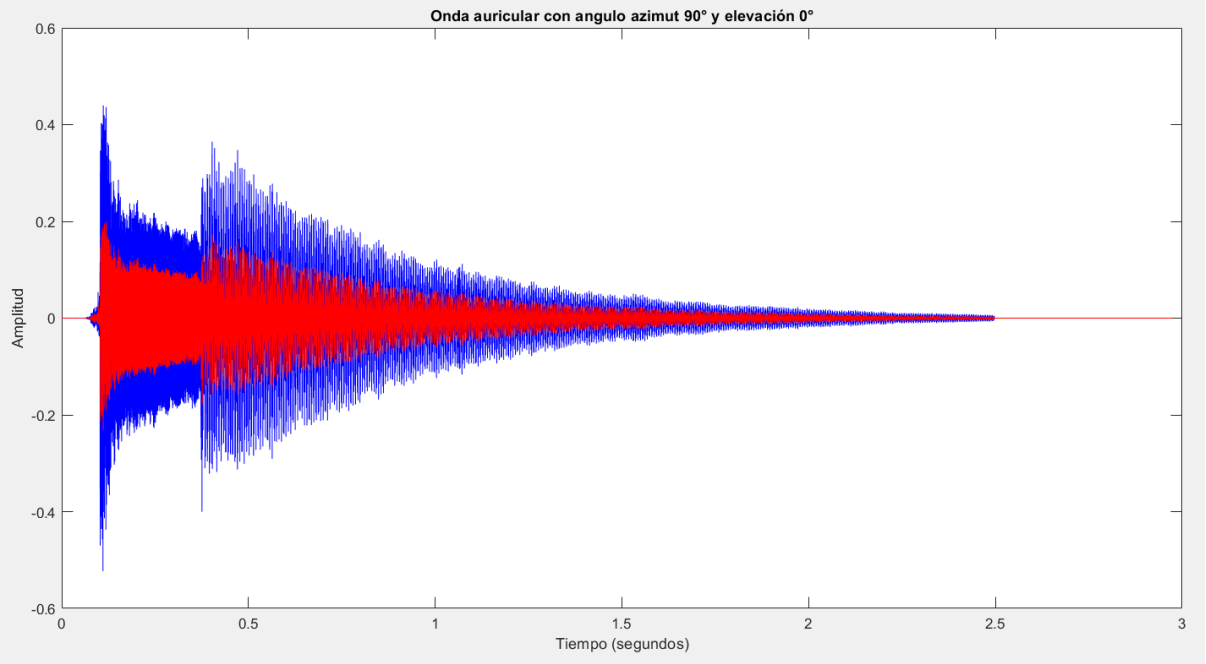
Nota. Onda del audio derecho con ángulo azimut 90° y elevación 0° en base a las muestras del MTI

Fuente: Realización propia (2021).

Una vez generada la transformada de Fourier, se realiza el producto punto entre la HRTF del MTI y el audio base. La señal resultante de este proceso se le extraerá la transformada inversa de Fourier, lo que generará una señal con el audio base. Este audio final obtiene las características particulares de procedencia de la HRTF del MTI, pero con el sonido del audio base.

Figura 32

Ondas Izquierda y derecha

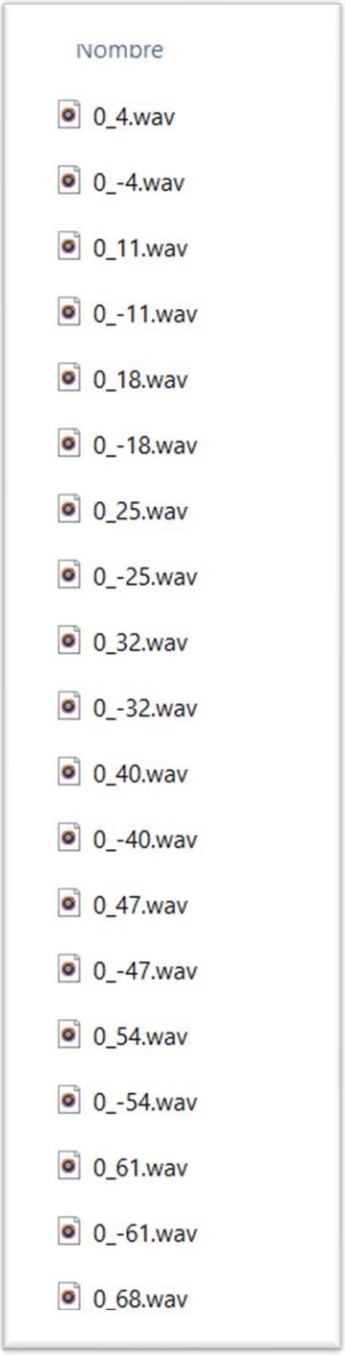


Nota. Ondas Izquierda (rojo) y derecha (azul) con ángulo azimut 90° y elevación 0°. Fuente: Realización propia (2021).

Realizado este procedimiento para todos los puntos de direcciones que se había generado previamente se tendrá un repositorio de varios audios '.wav'. Cada audio tendrá el mismo sonido con la diferencia que su dirección de procedencia será diferente. La forma en cómo se almacena estos audios para poder identificar sus ángulos es colocando la información en el nombre del archivo. El archivo tendrá la extensión '.wav', el nombre empieza con el ángulo de elevación, seguido de una raya baja y el ángulo de azimut. Como resultado se tendrá una carpeta de archivos de audio como se observa en la siguiente imagen figura 33.

Figura 33

Repositorio de audios



Fuente: Realización propia (2021).

3.2. IDENTIFICACIÓN DE CARACTERÍSTICAS PRINCIPALES PARA IDENTIFICACIÓN DE LA POSICIÓN DE LA FUENTE SONORA.

Una de las partes fundamentales para el presente proyecto es la extracción de las características de los audios. De la buena elección y forma de extraer las características dependerá en gran parte la eficacia que tenga el modelo que se generará. Haciendo referencia al termino “GIGO” en inglés (Garbage in, Garbage out) “basura entra, basura sale” en español, en el que básicamente si se ingresa datos erróneos o poco útiles al modelo se tendrá un resultado deficiente. La calidad y eficacia de este modelo depende primero de los datos que se extraigan y del algoritmo del modelo con el que se resuelva el problema.

El objetivo es encontrar las características que más información puedan aportar en base a la dirección de procedencia del sonido. Se mide cuanta información valiosa aporta una característica dependiendo de la correlación que tenga con la dirección de procedencia. En este caso lo que se sabe del audio acerca de la dirección de procedencia es el ángulo de azimut y de elevación. Por lo que, se sabe que una característica aporta información útil dependiendo de la correlación que tenga con estos ángulos.

En principio se extrae la mayor cantidad posible de características de los audios. A partir de aquí, se va depurando y seleccionando aquellas que se va a usar para el entrenamiento del modelo. Se aclara que el hecho de tener más características no significa tener más precisión o mayor eficacia en el modelo, incluso puede generar un efecto inverso. Sin embargo, esto tampoco quiere decir que se obtendrá cualquier tipo de características, sino que deben de estar relacionadas o en referencia a los conceptos vistos anteriormente en el marco teórico. En este sentido se toma en cuenta los términos como la intensidad, amplitud de onda, frecuencia del

sonido, tiempo del audio, periodo de la señal, etc. En este contexto las características que se deciden extraer son las siguientes:

- Nombre del audio.
- Ángulo de elevación.
- Ángulo de azimut.
- Zona de procedencia del sonido.
- Frecuencia del sonido.
- Intensidad Máxima.
- Intensidad Mínima.
- Intensidad Media.
- Intensidad máxima auricular derecho.
- Intensidad mínima auricular derecho.
- Intensidad media auricular derecho.
- Rango Intensidad auricular derecho.
- Intensidad máxima auricular izquierdo.
- Intensidad mínima auricular izquierdo.
- Intensidad media auricular izquierdo.
- Rango Intensidad auricular izquierdo.
- Diferencia de tiempo entre los picos de ambos auriculares.
- Diferencia de Intensidad entre los picos de ambos auriculares.
- ITD (Diferencia de tiempo interauricular)
- ILD (Diferencia de nivel interauricular)

Para la extracción de estas características se usa una iteración que recorre todos los audios que generados previamente. Adicional, se codifican funciones que extraerán cada una de estas características.

Nombre del audio:

Para el nombre del audio simplemente se usará la función Split con el separador '.' Que permitirá básicamente tomar el nombre del archivo, pero sin la extensión de este.

```
nombre=audio.split(".")[0]
```

Ángulo de elevación:

En cuanto al ángulo de elevación hay que recordar que este valor se lo colocó en el nombre del archivo. Para esto se usará nuevamente la función Split con el separador '_' y se toma el primer valor.

```
angElev=int(nombre.split("_")[0])
```

Ángulo de azimut:

En cuanto al ángulo de azimut se realiza el mismo procedimiento que para el ángulo de elevación, sólo que en este caso vamos a tomar el segundo valor.

```
angAzim=int(nombre.split("_")[1])
```

Zona:

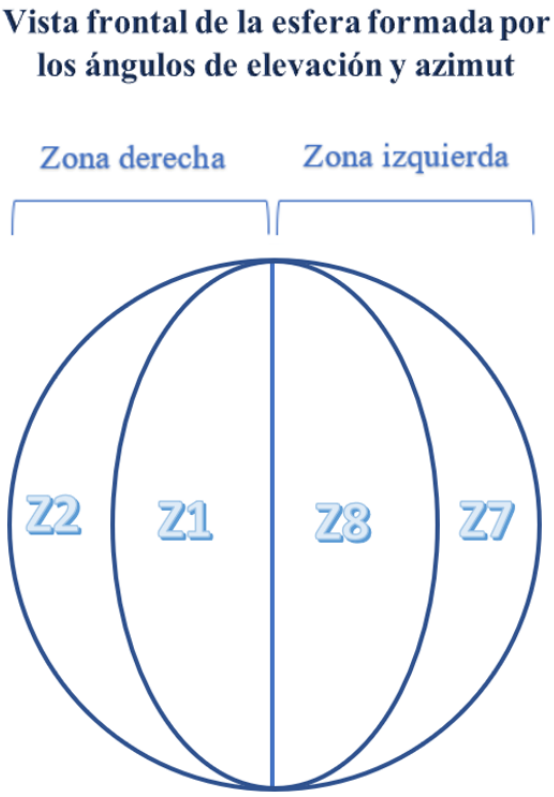
Dado que el modelo es de clasificación primero se divide o clasifica manualmente los diferentes audios del dataset según el ángulo de azimut en diferentes zonas. Para este caso los

audios se dividen en ocho zonas. La división se hace en base a la esfera generada por los ángulos, de este modo se identifican las siguientes zonas:

- (z1) Derecha-Frontal-Delante
- (z2) Derecha-Frontal-Lateral
- (z3) Derecha-Posterior-Lateral
- (z4) Derecha-Posterior-Detrás
- (z5) Izquierda-Posterior-Detrás
- (z6) Izquierda-Posterior-Lateral
- (z7) Izquierda-Frontal-Lateral
- (z8) Izquierda-Frontal-Delante

Figura 34

Vista frontal de la esfera formada por los ángulos de elevación y azimut



Fuente: Realización propia (2021).

Figura 35

Vista superior de la esfera formada por los ángulos de elevación y azimut

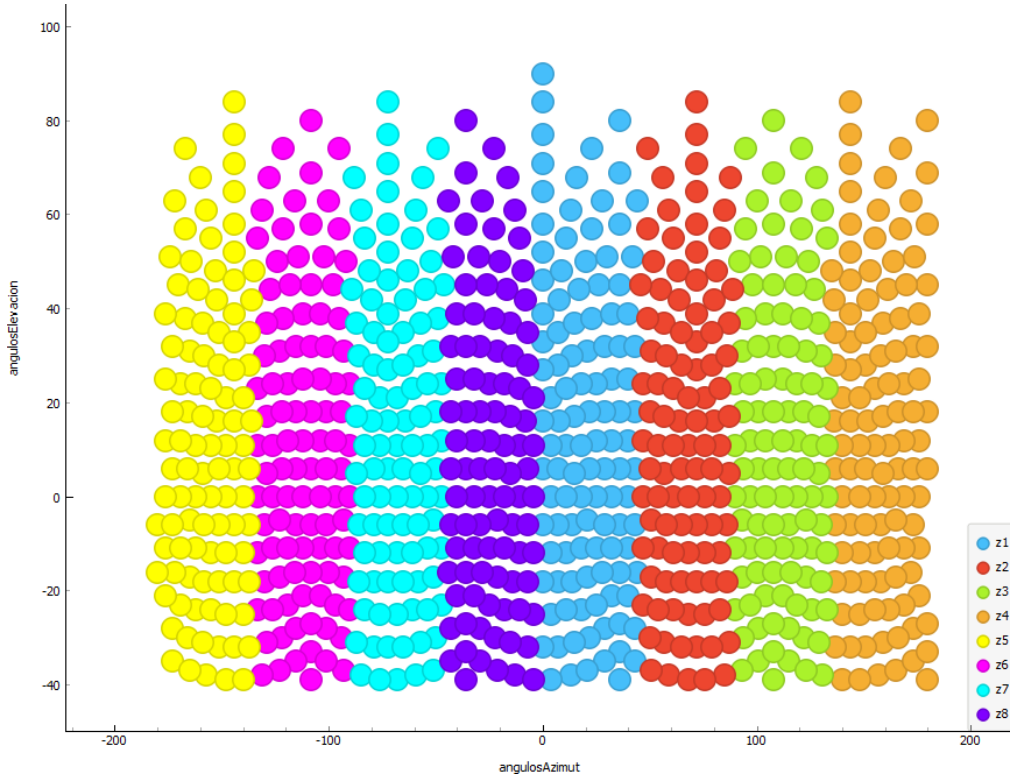


Fuente: Realización propia (2021).

Para entender de mejor manera la división de las zonas se debe comprender que se tiene la parte izquierda que representa los puntos con ángulo de azimut menor que cero grados y mayor o igual que -180 grados y la parte derecha que representa los puntos con ángulo de azimut mayor o igual que cero grados y menor o igual que 180 grados. Por último, se tiene una parte frontal que representa a los puntos con ángulo de azimut mayor o igual que -90 grados y menor que 90 grados y la parte posterior que representa los puntos con ángulo de azimut entre 90 y 180 grados, y entre -90 y -180 grados.

Figura 36

Imagen de la esfera "abierta"



Nota. Imagen de la esfera "abierta" dividida en las ocho zonas en base a los ángulos de azimut y de elevación

Fuente: Realización propia (2021).

Para la extracción de esta característica se usa la siguiente función y script:

```
zona=extraerZona(angElev,angAzim)

def extraerZona(angElev,angAzim):
    zona="z0"
    if (angAzim >=0 and angAzim <45):
        zona="z1"
    if (angAzim >=45 and angAzim <90):
        zona="z2"
    if (angAzim >=90 and angAzim <135):
        zona="z3"
    if (angAzim >=135 and angAzim <=180):
```

```
    zona="z4"
  if (angAzim >=-180 and angAzim <=-135):
    zona="z5"
  if (angAzim >-135 and angAzim <=-90):
    zona="z6"
  if (angAzim >-90 and angAzim <=-45):
    zona="z7"
  if (angAzim >-45 and angAzim <-0):
    zona="z8"
  return zona
```

Frecuencia del sonido:

La frecuencia del sonido se la obtiene al obtener la data del audio. Esto se obtiene con la función wavfile.read(). Para este caso como el audio base es el mismo, la frecuencia será la misma para todos por lo que no aportará información importante acerca de la procedencia del sonido. Sin embargo, la frecuencia ayudará a calcular otras características.

```
frecuencia, data = wavfile.read( ruta + audio )
```

Intensidad Máxima:

Para la extracción de la intensidad máxima se extraerá el pico más alto de la onda de ambos auriculares. Luego de tener estos dos valores nuevamente se extrae el máximo entre ambos que es el valor final de esta característica.

```
intMax=extraerIntensidadMaxima(data)

def extraerIntensidadMaxima(data):
  maximo=max([max(data[: ,0]),max(data[:,1])])
  return máximo
```

Intensidad Mínima:

Para la extracción de la intensidad mínima se extrae el valor más bajo de la onda de ambos auriculares. Luego de tener estos dos valores nuevamente se extrae el mínimo valor de estos dos que es el valor de esta característica.

```
intMin=extraerIntensidadMinima(data)
```

```
def extraerIntensidadMinima(data):
    minimo=min([min(data[:,0]),min(data[:,1])])
    return minimo
```

Intensidad Media:

En el caso de la intensidad media se hace una suma entre la intensidad máxima e intensidad mínima.

```
intMed=(intMax+intMin)/2
```

Intensidad máxima auricular derecho:

Para la intensidad máxima del auricular derecho se hace uso de la función max() y se coloca como parámetro la data del lado derecho.

```
maxD=float(max(data[:,1]))
```

Intensidad mínima auricular derecho:

Para la intensidad mínima del auricular derecho se hace uso de la función min() y se coloca como parámetro la data del lado derecho.

```
minD=float(min(data[:,1]))
```

Intensidad media auricular derecho:

Para extraer esta característica se saca un promedio entre la intensidad máxima e intensidad mínima del auricular derecho.

```
medD=(maxD+minD)/2
```

Rango Intensidad auricular derecho:

El rango de intensidad del auricular derecho es la diferencia entre la intensidad máxima menos la intensidad mínima del auricular derecho.

```
amplitudD=maxD-minD
```

Intensidad máxima auricular izquierdo:

Para la intensidad máxima del auricular izquierdo se hace uso de la función max() y se coloca como parámetro la data del lado izquierdo.

```
maxI=float(max(data[:,0]))
```

Intensidad mínima auricular izquierdo:

Para la intensidad mínima del auricular izquierdo se hace uso de la función min() y se coloca como parámetro la data del lado izquierdo.

```
minI=float(min(data[:,0]))
```

Intensidad media auricular izquierdo:

Para extraer esta característica se saca un promedio entre la intensidad máxima e intensidad mínima del auricular izquierdo.

```
medI=(maxI+minI)/2
```

Rango Intensidad auricular izquierdo:

El rango de intensidad del auricular izquierdo es la diferencia entre la intensidad máxima menos la intensidad mínima del auricular izquierdo.

```
amplitudI=maxI-minI
```

Diferencia de tiempo entre los picos de ambos auriculares:

La diferencia que existe entre los picos es una característica que puede aportar información relevante acerca de la dirección de procedencia del sonido. Esto debido a que tiene relación al concepto de ITD. Los picos de ambas ondas representarían la intensidad máxima del sonido por lo que al sacar la diferencia entre estos picos podemos medir el retardo de llegada del sonido entre ambos auriculares. Esta característica se la obtendrá en segundos ya que al extraer la diferencia se estará sacando la diferencia en índices, por lo que al dividirla por la frecuencia dará la diferencia en segundos.

```
difMax=extraerDiferenciaMaximos(data,frecuencia)

def extraerDiferenciaMaximos(data,frecuencia):
    diferencia=(list(data[:,0]).index(max(data[:,0]))-
    list(data[:,1]).index(max(data[:,1])))/frecuencia
    return diferencia
```

Diferencia de Intensidad entre los picos de ambos auriculares:

La diferencia entre la intensidad de los picos de ambos auriculares también parte en base al ILD. Esta característica también puede aportar información para localizar la dirección de la fuente del sonido. Esto debido a que el sonido al estar más cerca de uno u otro tendrá una

intensidad más alta en ese auricular. Del mismo modo el sonido al estar más lejos de un auricular, este tendrá un pico de onda menor al auricular que está más cerca de la fuente del sonido.

```
difMaxNivel=extraerDiferenciaMaximosNiveles(data)


def extraerDiferenciaMaximosNiveles(data):
   diferencia=max(data[:,0])-max(data[:,1])
   return diferencia
```

ITD (Diferencia de tiempo interauricular):

Esta característica como ya se ha visto anteriormente es de las más importantes. La información que aporta es esencial debido a que el auricular que esté más cerca de la fuente de sonido será el primero que receptará la onda del sonido. De esta manera mientras más diferencia exista entre la llegada de la onda para cada auricular ayudará a determinar el ángulo de azimut de procedencia del sonido. Para extraer esta característica se usa correlación cruzada.

Correlación Cruzada:

En este caso se utiliza una correlación cruzada de dos matrices N-dimensionales, la salida es la correlación cruzada lineal discreta completa de las entradas. Es decir, una matriz N-dimensional que contiene un subconjunto de la correlación cruzada lineal discreta de los datos del auricular derecho con los datos del auricular izquierdo.

La correlación z de dos matrices d-dimensionales x e y se define como:

```
z[...,k,...] = sum[..., i_l, ...] x[..., i_l,...] * conj(y[..., i_l - k,...])
```

$$z[k] = (x * y)(k - N + 1) = \sum_{l=0}^{||x||-1} x_l y^*_{l-k+N-1}$$

$$k = 0, 1, \ldots, ||x|| + ||y|| - 2$$

La manera en cómo funciona la relación cruzada que se usa en la extracción de esta característica es multiplicando los datos de la onda de sonido del auricular izquierdo con los datos de la onda del auricular derecho sucesivamente posición por posición de principio a fin por cada onda. Para explicar de una manera más clara lo anterior se usa el ejemplo de un sonido que se lo adecuará para que luzca que su dirección de procedencia sea del lado derecho.

Se procede a simular dos ondas de sonido con los siguientes datos:

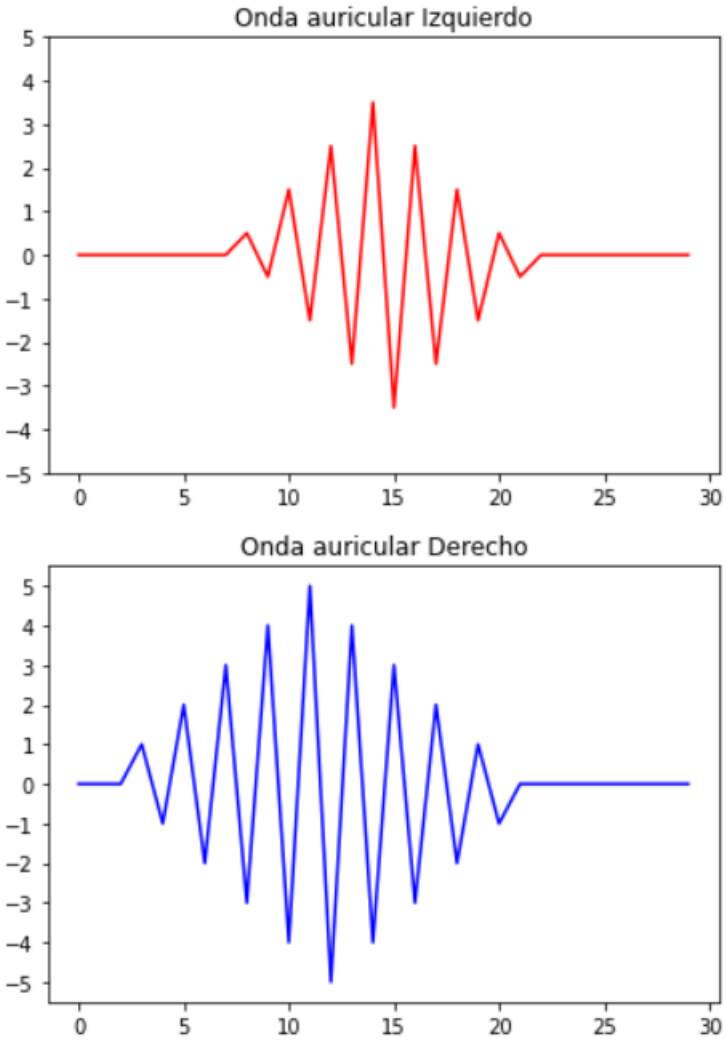
```
ondaDerecha=[0,0,0,1,-1,2,-2,3,-3,4,-4,5,-5,4,-4,3,-3,2,-2,1,-
1,0,0,0,0,0,0,0,0,0]
```

```
ondaIzquierda=[0,0,0,0,0,0,0,0,0.5,-0.5,1.5,-1.5,2.5,-2.5,3.5,-3.5,2.5,-
2.5,1.5,-1.5,0.5,-0.5,0,0,0,0,0,0,0,0]
```

De manera gráfica se muestra de la siguiente manera:

Figura 37

Ondas auricular izquierdo y auricular derecho



Fuente: Realización propia (2021).

Datos Auricular Izquierdo

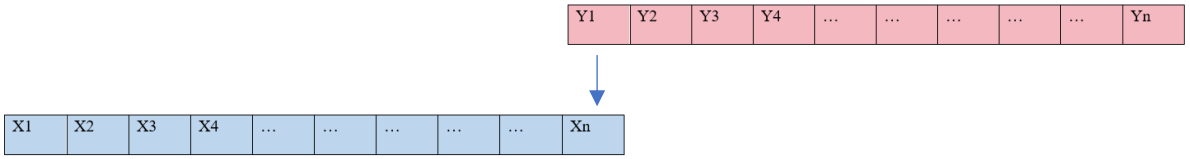
Y1	Y2	Y3	Y4	…	…	…	…	…	Yn

Datos Auricular Derecho

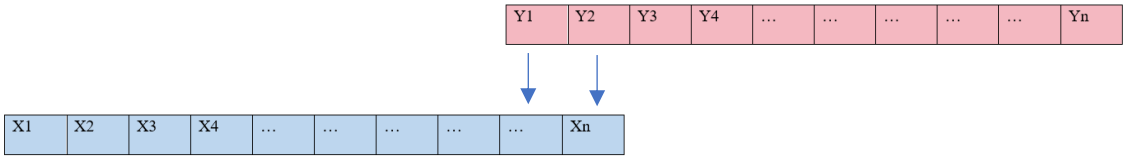
X1	X2	X3	X4	…	…	…	…	…	Xn

Representación gráfica de la correlación cruzada

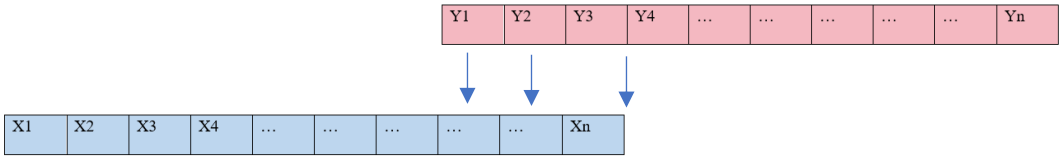
Valor 1 = Sumatoria de la multiplicación de los siguientes datos:



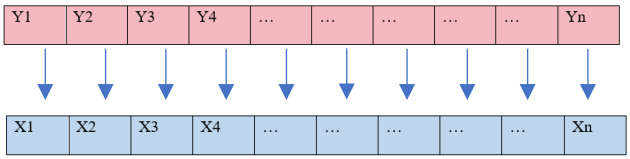
Valor 2 = Sumatoria de la multiplicación de los siguientes datos:



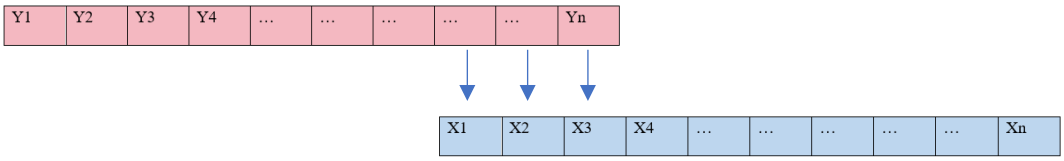
Valor 3 = Sumatoria de la multiplicación de los siguientes datos:



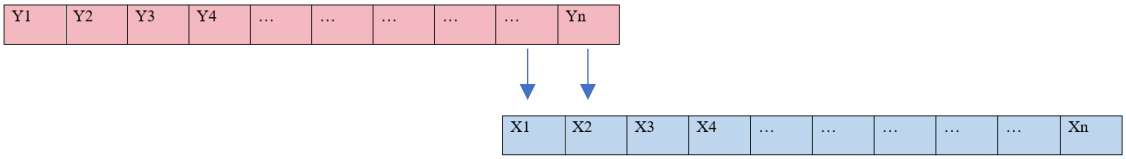
Valor n= Sumatoria de la multiplicación de los siguientes datos:



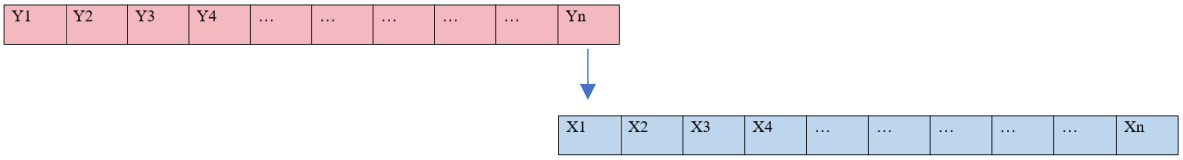
Valor (n*2)-3= Sumatoria de la multiplicación de los siguientes datos:



Valor (n*2)-2= Sumatoria de la multiplicación de los siguientes datos:



Valor (n*2)-1= Sumatoria de la multiplicación de los siguientes datos:



El resultado de la correlación cruzada será un vector de tamaño n*2 – 1, donde n corresponde a la cantidad de datos del auricular izquierdo y a la vez del auricular derecho.

El objetivo de utilizar la correlación cruzada es que, del vector resultante al tomar el dato mayor, la posición de este representará el punto en donde las ondas coinciden. Teniendo este índice hay que mirar la diferencia que existe desde este índice al índice del medio, es decir del “valor n”. El valor del medio es la correlación que existe cuando los datos de ambas ondas

coinciden en los índices. En la siguiente figura se muestra los vectores correlación entre los auriculares derecho e izquierdo:

Figura 38

Correlación entre los auriculares derecho e izquierdo



Fuente: Realización propia (2021).



Fuente: Realización propia (2021).

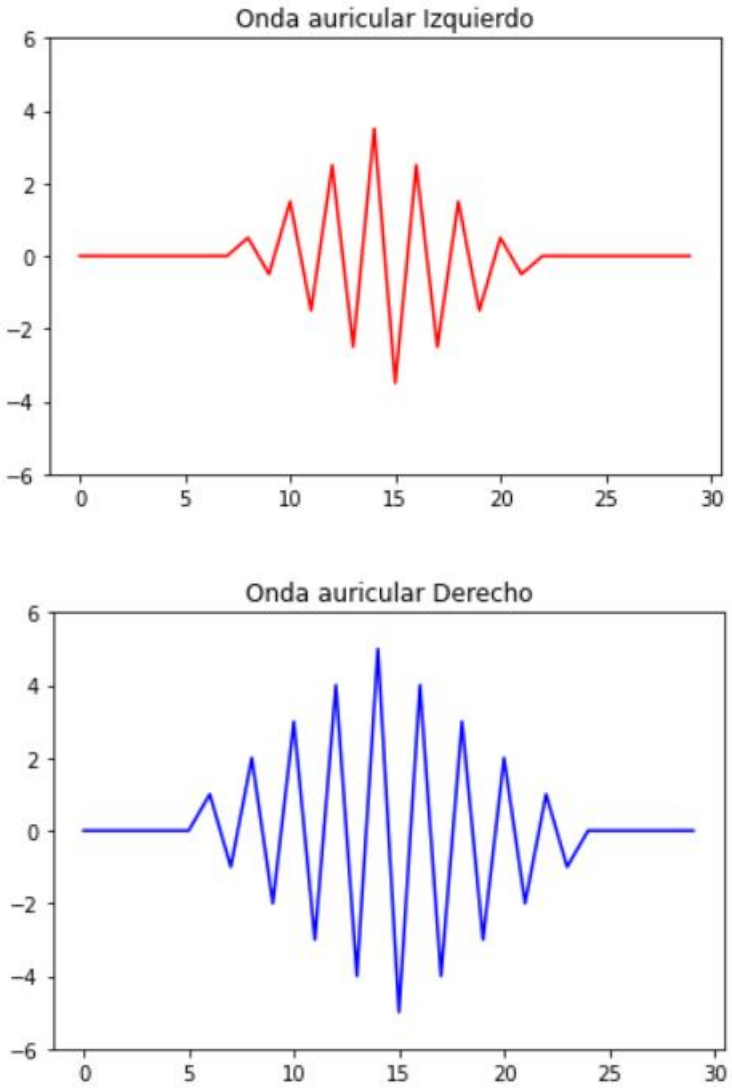
Como se puede observar la diferencia entre ambos gráficos es que son inversos en base a la mitad de la longitud de todos los datos (línea roja). También se puede observar que existe un pico máximo que para el primer gráfico se encuentra en el índice 32 y para el segundo gráfico se encuentra en el índice 26. En este sentido hay que tomar en cuenta que la mitad se encuentra en el índice 29, si se saca la diferencia absoluta entre ambos picos máximos se puede encontrar que se tiene una diferencia de 3 índices. De esta forma hemos extraído el retardo entre ambos auriculares, ya que como el punto medio es cuando no existe ningún retardo entre las ondas.

Para comprobarlo se ha retrasado tres índices la onda de sonido del auricular derecho, quedando los datos de la siguiente manera:

```
ondaDerecha=[0,0,0,0,0,0,1,-1,2,-2,3,-3,4,-4,5,-5,4,-4,3,-3,2,-2,1,-
1,0,0,0,0,0,0]
ondaIzquierda=[0,0,0,0,0,0,0,0,0.5,-0.5,1.5,-1.5,2.5,-2.5,3.5,-3.5,2.5,-
2.5,1.5,-1.5,0.5,-0.5,0,0,0,0,0,0,0,0]
```

Figura 39

Ondas auriculares izquierdo y derecho

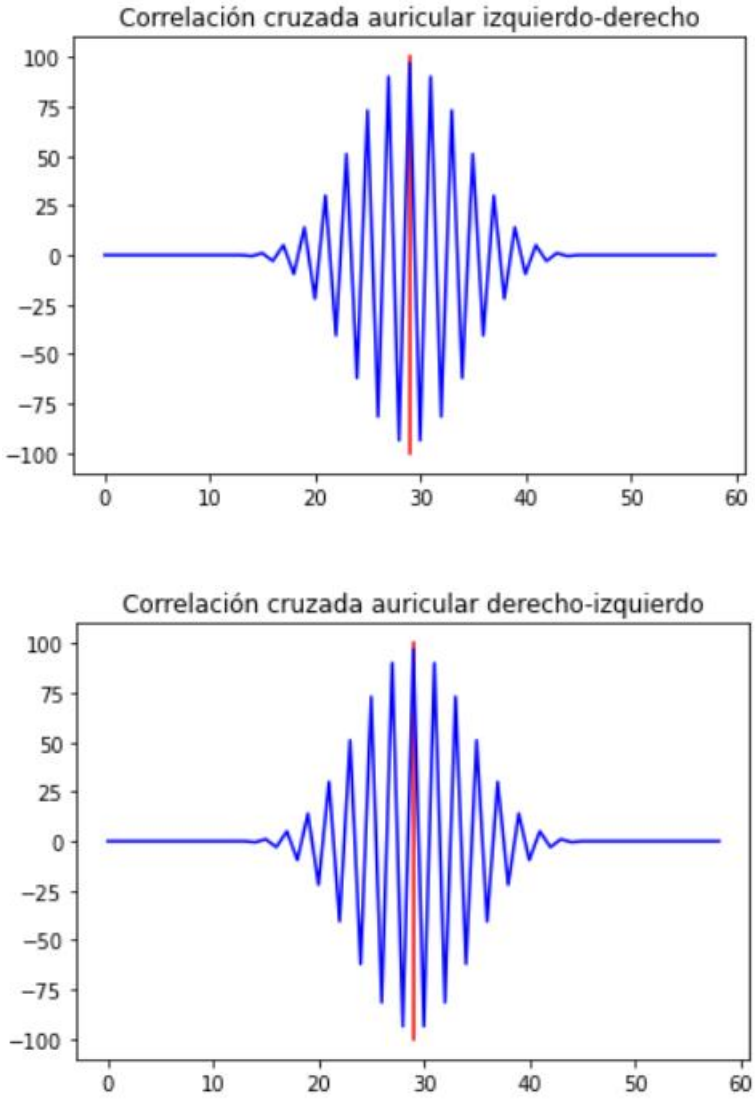


Fuente: Realización propia (2021).

A primera vista se puede observar que las ondas son simétricas entre sí después de haber eliminado el retraso. Sin embargo, es necesario observar el vector de correlación cruzada que existe entre ambos auriculares para estar seguros de que las ondas no tienen retardo.

Figura 40

Correlación cruzada



Fuente: Realización propia (2021).

Como se puede observar al eliminar el retraso entre las ondas se tiene los mismos vectores de correlación cruzada y de igual manera el pico máximo se encuentra en el centro.

```
itd=extraerITD(data,frecuencia)


def extraerITD(data,frecuencia):
    ma=max(max(data[:,0]),max(data[:,1]))
    y1=data[:,0]/ma
    y2=data[:,1]/ma
    corr=signal.correlate(y1, y2)
    mitad=len(data[:,0])-1
    diferencia=np.argmax(corr)-mitad
    return diferencia/frecuencia
```

ILD (Diferencia de nivel interauricular):

El ILD es otra de las características más importantes. Esto ya que aporta información de la dirección de procedencia del sonido. Si un auricular tiene mayor nivel que el otro significa que el sonido proviene de esa dirección. Por lo que el ILD debe tener una correlación directa con el ángulo de azimut. Para esto se extrae la diferencia entre la presión sonora entre ambos auriculares.

"La presión sonora se define como la desviación instantánea de la presión con respecto a la presión ambiente (atmosférica en nuestro caso) causada por una onda sonora. La unidad de medida internacional es el Pascal (Pa). El margen de frecuencias audibles por el humano es de 20 Hz a 20 kHz (las frecuencias superiores son conocidas como ultrasónicas y las inferiores como infrasónicas) y el nivel de presión audible abarca desde los 20 µPa (tomado como valor de referencia, umbral de audición) hasta 200 Pa (umbral de dolor). Este rango es muy amplio (ocupa 7 órdenes de magnitud), lo que hace necesario el uso de una escala logarítmica. Por ello la magnitud pasa a llamarse nivel de presión sonora (NPS), también conocida en inglés como Sound Pressure Level (SPL), la cual se define como:" (GANIC, 2020)

$$NPS(dB) = 20 \cdot \log \frac{P_1}{Pref}$$

Donde P1 es la presión sonora instantánea y Pref es la presión usada como referencia (20 µPa). (GANIC, 2020)

```
ild=extraerILD(data)


def extraerILD(data):
    solucion1=10*math.log((sum(data[:,0]**2)/len(data[:,0]))/(0.00002**2))
    solucion2=10*math.log((sum(data[:,1]**2)/len(data[:,1]))/(0.00002**2))
    diferencia=solucion1-solucion2
    return diferencia
```

3.3. DEFINICIÓN DEL MÉTODO DE MACHINE LEARNING ÓPTIMO PARA CLASIFICACIÓN DE ESTE TIPO DE DATOS.

Según lo revisado previamente se puede dar cuenta que el entrenamiento óptimo para la clasificación de los diferentes audios según su dirección de procedencia es utilizar un entrenamiento supervisado. Esto dado que se tiene los features (características) con su respectivo target (la zona de la que proviene el sonido). De esta forma se da por descartado el uso de modelos como los clustering, k-Medias, clusterización jerárquica, Density Based Scan Clustering (DBSCAN), Modelo de Agrupamiento Gaussiano, entre otros. Dado que estos se basan en aprendizaje no supervisado.

Una vez revisado el tipo de entrenamiento del que se hace uso, se procede a definir el método óptimo. Hay que entender que para este problema en específico hay tres modelos de entrenamiento que pueden funcionar bien dado las características del modelo. Estos modelos son:

KN-Neigborhts: Este modelo puede funcionar bien para el modelo, ya que se enfoca en ver las similitudes con cada característica permitiendo ubicar el audio en la zona cuyas

características se asemejen. En este caso el inconveniente principal sería que a primera vista algunas zonas comparten características en común por el hecho de estar ubicadas en la misma zona ya sea superior e inferior, frontal y posterior o derecha e izquierda (figura 41). Esto puede generar que el modelo llegue a confundirse entre zonas principalmente cercanas.

Figura 41

Características de la zona



Nota. Imagen que muestra como la zona uno puede compartir características con la zona dos por estar en la zona frontal. De igual manera la zona uno puede compartir características con la zona cuatro por estar en la zona derecha. Fuente: Realización propia (2021).

Modelo Random Forest: Este modelo funcionaría bien con este proyecto. En este caso el modelo se vuelve un poco más complejo por el hecho de que el problema se resuelve por una

sucesión de decisiones. Al ser un conjunto de árboles de decisión, por cada árbol tendremos un output. En caso el modelo crearía varias estructuras de control de decisión dependiendo de cada una de las variables o features que le aportemos.

Redes neuronales: Este es el modelo más complejo de todos. Esto debido que por cada neurona tendremos varias entradas, varias salidas y varias capas según como se la programe. Sin embargo, esto permite que generalmente se tenga una mayor precisión.

Una vez visto los diferentes modelos que se utilizará se pasa al método óptimo para entrenar el modelo. Entendiendo el contexto del manejo de datos, el método óptimo debe iniciar con la limpieza del dataset, es decir analizar los datos y prepararlos para poder empezar con el entrenamiento. Una vez que se tiene el dataset limpio y ordenado se revisará que características servirán para entrenar el modelo. En este caso se toma en cuenta la correlación que se tenga en base a los ángulos de elevación y azimut ya que estos son los que indican la dirección del sonido. Inmediatamente seleccionadas las características o features que servirán para el entrenamiento se empieza el entrenamiento con cada uno de los modelos presentados previamente. La idea es evaluar el rendimiento en cada uno de los modelos y de esta forma ir depurando los datos o realizando cambios en los parámetros de cada uno de estos modelos. Al final se tomará el modelo que mayor precisión dé en base a los datos de validación o ‘test data’.

3.4. DEFINICIÓN DE LA HERRAMIENTA DE PROCESAMIENTO ACÚSTICO Y MACHINE LEARNING

En cuanto a las herramientas que se utilizan para el procesamiento acústico y de machine learning se toma en cuenta aspectos como el conocimiento y el manejo previo que se tenga en

esta herramienta, la potencia de procesamiento y las herramientas internas que permitan facilitar la creación del modelo para este proyecto.

HERRAMIENTAS PARA EL PROCESAMIENTO ACÚSTICO:

MATLAB: Es un entorno de desarrollo integrado. Este tiene su propio lenguaje de programación. Este software aporta un sistema de cómputo numérico. De hecho, su punto fuerte es el cálculo matemático.

"MATLAB ® combina un entorno de escritorio ajustado para procesos iterativos de análisis y diseño con un lenguaje de programación que expresa directamente las matemáticas de matrices y arreglos. Incluye Live Editor para crear scripts que combinan código, salida y texto formateado en un cuaderno ejecutable." (MATLAB, 2022)

En este caso Matlab es la opción más adecuada debido a que es bastante potente con el cálculo matemático, en este caso realizando operaciones con las matrices que contienen los datos de los audios. De este modo permite calcular las diferentes operaciones para obtener el dataset inicial de audios necesario para el entrenamiento.

Con esta herramienta se tiene el beneficio de que se ha tenido una interacción previa y se tiene gran parte de conocimiento acerca de las capacidades del software. A demás, del conocimiento de su lenguaje de programación. Este aspecto es importante porque reducirá el tiempo que se tome en la curva de aprendizaje para el desarrollo en esta aplicación para el procesamiento acústico.

Otra ventaja de esta herramienta es que viene con librerías preinstaladas aparte de otras que permite ahorrar la escritura de código innecesaria.

HERRAMIENTAS PARA MACHINE LEARNING:

Python: Es un lenguaje de programación de alto nivel interpretado. Este lenguaje es multipropósito ya que se desempeña bastante bien en varias áreas. Junto con el lenguaje R es un ícono en la ciencia de datos y el machine learning. En este caso Python se lo utiliza para el procesamiento de los datos y la obtención de las características de cada audio. Se hace uso de funciones y operaciones matemáticas para extraer las características. Además, se hace uso de la potencia de Python en el manejo de datos para poder crear el dataframe de los audios con sus features. El uso de este lenguaje se vuelve sencillo por el hecho de que se tiene bastante experiencia con varios proyectos y la escritura de código con Python.

Orange: Este es un software dedicado a la minería de datos y el análisis predictivo. Este programa se lo puede instalar y manipular desde Anaconda que es una distribución libre de Python y R. Orange aporta un entorno gráfico de Drag-and-Drop (Arrastrar y soltar) lo que facilita bastante el flujo y manipulación de los datos además del manejo de la variación de los parámetros de cada modelo. Este software es el que se utiliza para el entrenamiento y las pruebas. Orange es un software relativamente nuevo en cuanto a la experiencia previa con este, ya que es la primera vez que se lo utilizará. Sin embargo, los conceptos de los modelos y del entrenamiento si es información previamente aprendida, por lo que no es difícil el adaptarse a esta herramienta. De esta forma este software es el más apropiado para crear el modelo final.

3.5. LIBRERÍAS Y CLASES DE LA HERRAMIENTA PARA PROCESAMIENTO ACÚSTICO Y PARA CLASIFICACIÓN DE SEÑALES

En el caso de Matlab, este software contiene una amplia colección de funciones. Una gran ventaja de esta herramienta es que con la instalación inicial se tiene por defecto bastantes funciones que para el caso de este proyecto son suficientes. Entre estas funciones para el procesamiento del audio existen funciones para obtener la matriz de datos de los diferentes audios. De igual manera a partir de una matriz permite generar un audio. Entre otras funciones están las que permiten realizar gráficos. Estas funciones permiten graficar las ondas de los audios o generar gráficos en tres dimensiones necesarios para generar la esfera formada por los ángulos de azimut y elevación. Entre las principales funciones se tiene:

```
[azimuth,elevation,r] = cart2sph(x,y,z)

[x,y,z] = sph2cart(azimuth,elevation,r)

[y,Fs] = audioread(filename)

Y = fft(X)

X = ifft(Y)

scatter3(X,Y,Z)

plot(X,Y)

sound(y,Fs)
```

En el caso de Python se utilizan varias librerías tanto para el procesamiento de datos como para los cálculos de las diferentes características de los audios. Entre las principales librerías son:

NumPy: Esta librería se enfoca en el cálculo matemático de matrices. Es bastante potente con las matrices N-Dimensionales en base a los conceptos de vectorización, indexación y transmisión.

Pandas: esta librería se especializa en el manejo y el análisis de estructura de datos. Permite generar nuevas estructuras de datos basadas en arrays de la librería NumPy incluyendo algunas funcionalidades. Puede obtener datos directamente de ficheros con formato CSV, Excel e incluso bases de datos. El acceso a los datos se lo realiza mediante índices, o nombres tanto para filas o columnas. Sus estructuras principales son Series y Dataframe.

Matplotlib: Es una librería que permite generar gráficos tanto para visualizaciones estáticas, animadas e interactivas. Es bastante intuitiva y fácil de utilizar. De igual manera permite realizar una importación de los gráficos en diferentes formatos de archivo.

3.6.REQUERIMIENTOS FUNCIONALES

En cuanto a los requerimientos funcionales de este proyecto se han definido los siguientes:

- RF1: El modelo debe dar como resultado la zona de procedencia del sonido.
- RF2: El modelo debe tener una precisión de al menos un 70%.
- RF3: El modelo debe tomar como input características de un audio biauricular.
- RF4: El modelo debe responder (en los términos descritos por RF2) a inputs que no se usaron durante el proceso de entrenamiento.

4. IMPLEMENTACIÓN DEL MODELO

4.1.DEFINICIÓN DE LA ARQUITECTURA DEL MODELO

El modelo toma como entrada el dataset de las características de los diferentes audios cuyas direcciones fuente son distintas. En este caso el dataset viene en formato de archivo CSV. De esta forma dentro de la herramienta de Orange se importa el archivo ‘datosData1.csv’ que es el que contiene los datos para entrenamiento. El dataset contiene 826 registros, cada uno de una dirección diferente.

Figura 42

Formato de archivo CSV



Fuente: Realización propia (2021).

Figura 43

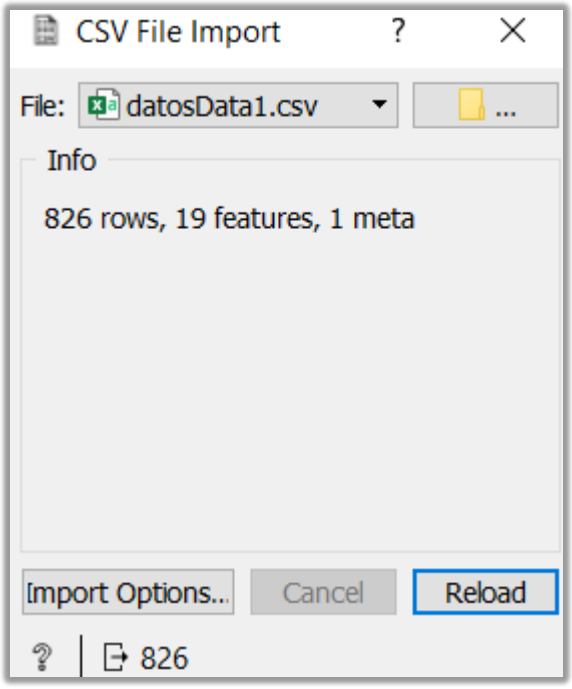
Muestra de los primeros 22 registros del dataset

	nombres	angulosElev	angulosAzim	zonas	frecuencias	intensidadM	intensidadM	intensidadM	intensidadM	intensidadM	amplitudDer	intensidadM	intensidadM	intensidadM	amplitudIzqu	diferenciaM	diferenciaM	itd	ild
2	-11_-104	-11	-104	z7	44100	14519	-16168	-824.5	4371.0	-4337.0	17.0	8708.0	14519.0	-16168.0	-824.5	30687.0	-0.28428571	10148	-0.00070294	-5.144.923.155.571.510
3	-11_-112	-11	-112	z7	44100	14193	-15929	-868.0	4208.0	-4158.0	25.0	8366.0	14193.0	-15929.0	-868.0	30122.0	0.015034013	9985	-0.00068027	-46.033.343.018.983.700
4	-11_-119	-11	-119	z7	44100	14029	-15591	-781.0	4589.0	-4653.0	-32.0	9242.0	14029.0	-15591.0	-781.0	29620.0	0.015079365	9440	0.005963718	-5.499.461.604.669.980
5	-11_-126	-11	-126	z7	44100	13731	-15383	-826.0	5099.0	-5816.0	-358.5	10915.0	13731.0	-15383.0	-826.0	29114.0	0.007437641	8632	0.006090707	-47.245.549.892.064.100
6	-11_-162	-11	-162	z7	44100	11400	-13673	-1136.5	7477.0	-9231.0	-877.0	16708.0	11400.0	-13673.0	-1136.5	25073.0	0.009047619	3923	-0.00015873	-2.623.045.056.089.830
7	-11_-169	-11	-169	z7	44100	10815	-13022	-1103.5	8264.0	-10129.0	-932.5	18393.0	10815.0	-13022.0	-1103.5	23837.0	-9,07E+10	2551	-9,07E+10	-18.601.889.191.548.900
8	-11_-176	-11	-176	z7	44100	10058	-12223	-1082.5	9184.0	-11239.0	-1027.5	20423.0	10058.0	-12223.0	-1082.5	22281.0	-2,27E+11	874	-2,27E+11	-0.290387897181 0059
9	-11_-18	-11	-18	z8	44100	14078	-16187	-1054.5	8299.0	-10518.0	-1109.5	18817.0	14078.0	-16187.0	-1054.5	30265.0	-0.00412698	5779	-0.00015873	-20.786.547.262.412.100
10	-11_-25	-11	-25	z8	44100	15084	-17019	-967.5	7493.0	-9748.0	-1127.5	17241.0	15084.0	-17019.0	-967.5	32103.0	-0.00417233	7591	-0.00024943	-3.552.125.838.652.720
11	-11_-32	-11	-32	z8	44100	15257	-17255	-999.0	6701.0	-8876.0	-1087.5	15577.0	15257.0	-17255.0	-999.0	32512.0	-0.00424036	8556	-0.00031746	-3.406.584.278.530.940
12	-11_-40	-11	-40	z8	44100	15422	-17456	-1017.0	6131.0	-8124.0	-996.5	14255.0	15422.0	-17456.0	-1017.0	32878.0	-0.00034013	9291	-0.00038548	-4.328.651.608.592.500
13	-11_-47	-11	-47	z8	44100	15364	-17568	-1102.0	5600.0	-6972.0	-686.0	12572.0	15364.0	-17568.0	-1102.0	32932.0	0.003333333	9764	-0.00045351	-4.004.955.247.976.620
14	-11_-54	-11	-54	z8	44100	15445	-17487	-1021.0	5269.0	-6068.0	-399.5	11337.0	15445.0	-17487.0	-1021.0	32932.0	-0.00043083	10176	-0.00052154	-5.214.866.355.280.370
15	-11_-90	-11	-90	z8	44100	14626	-16676	-1025.0	4939.0	-5542.0	-301.5	10481.0	14626.0	-16676.0	-1025.0	31302.0	-0.28428571	9687	-0.00070294	-5.616.693.508.736.600
16	-11_-97	-11	-97	z7	44100	14550	-16488	-969.0	4751.0	-4725.0	13.0	9476.0	14550.0	-16488.0	-969.0	31038.0	-0.28428571	9799	-0.00070294	-6.117.674.832.903.780
17	-11_104	-11	104	z6	44100	14519	-16168	-824.5	14519.0	-16168.0	-824.5	30687.0	4371.0	-4337.0	17.0	8708.0	0.284285714	-10148	0.000702947	5.144.923.155.571.510
18	-11_112	-11	112	z6	44100	14193	-15929	-868.0	14193.0	-15929.0	-868.0	30122.0	4208.0	-4158.0	25.0	8366.0	-0.01503401	-9985	0.000680272	46.033.343.018.983.700
19	-11_119	-11	119	z6	44100	14029	-15591	-781.0	14029.0	-15591.0	-781.0	29620.0	4589.0	-4653.0	-32.0	9242.0	-0.01507936	-9440	-0.00596371	5.499.461.604.669.980
20	-11_126	-11	126	z6	44100	13731	-15383	-826.0	13731.0	-15383.0	-826.0	29114.0	5099.0	-5816.0	-358.5	10915.0	-0.00743764	-8632	-0.00600907	47.245.549.892.064.100
21	-11_162	-11	162	z6	44100	11400	-13673	-1136.5	11400.0	-13673.0	-1136.5	25073.0	7477.0	-9231.0	-877.0	16708.0	-0.00904761	-3923	0.000158730	2.623.045.056.089.830
22	-11_169	-11	169	z6	44100	10815	-13022	-1103.5	10815.0	-13022.0	-1103.5	23837.0	8264.0	-10129.0	-932.5	18393.0	9,07E+10	-2551	9,07E+10	18.601.889.191.548.900

Fuente: Realización propia (2021).

Figura 44

Herramienta de Orange

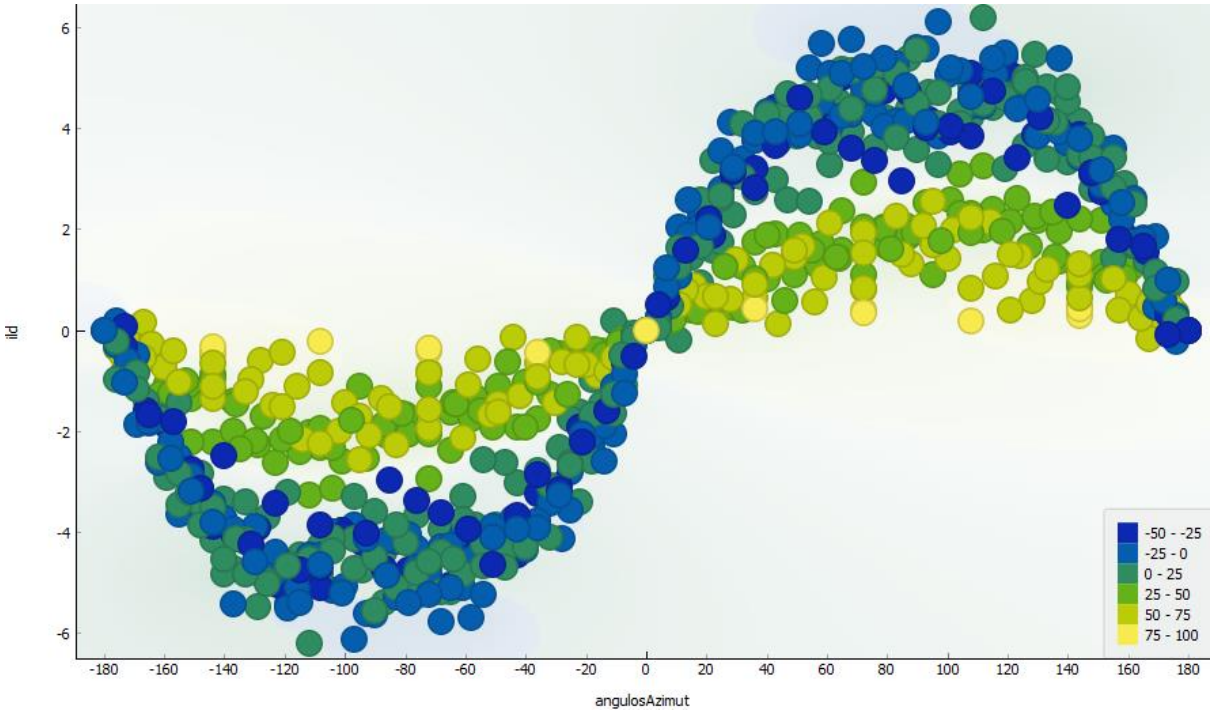


Fuente: Realización propia (2021).

Una vez importado los datos a la herramienta, es importante tener una visualización gráfica de estos. De esta manera se revisa las correlaciones de forma visual de las diferentes características con los ángulos de azimut y de elevación que son los que determinan la dirección de procedencia del audio. En este caso las más importantes para la revisión serían el ILD y el ITD.

Figura 45

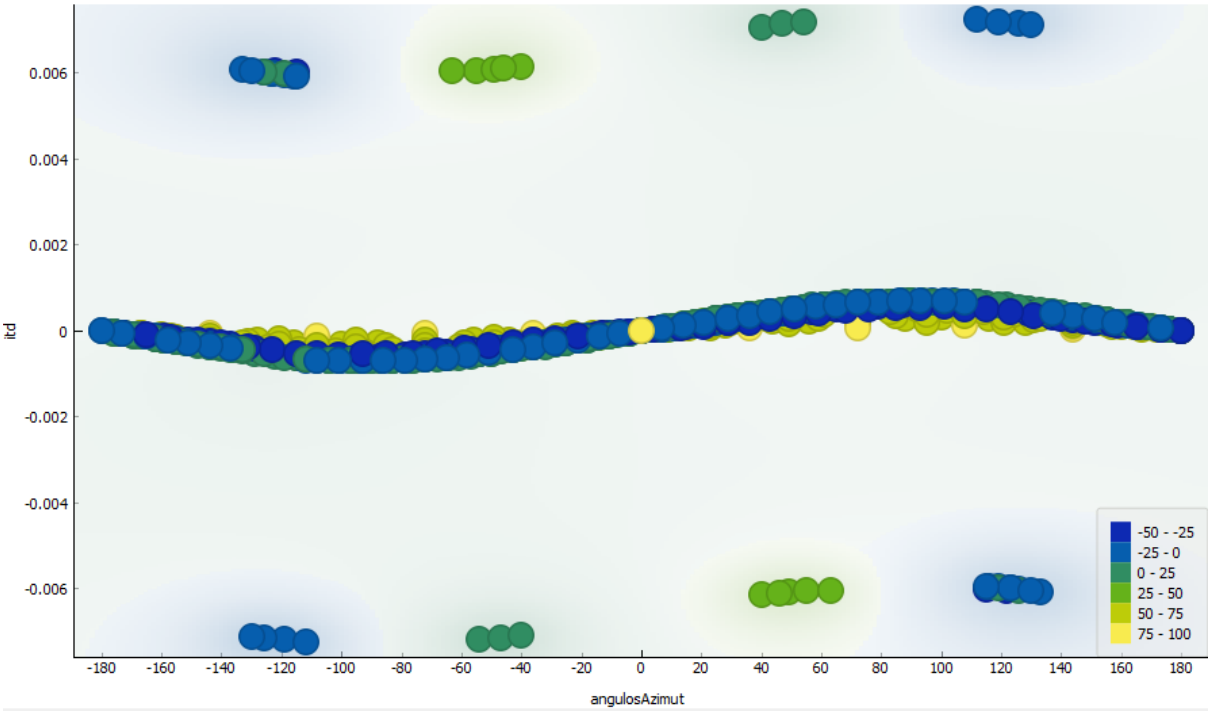
Gráfico de dispersión



Nota. Gráfico de dispersión con valores del ILD vs ángulos Azimut, con una diferencia de color según la característica ángulos Elevación. Fuente: Realización propia (2021).

Figura 46

Gráfico de dispersión

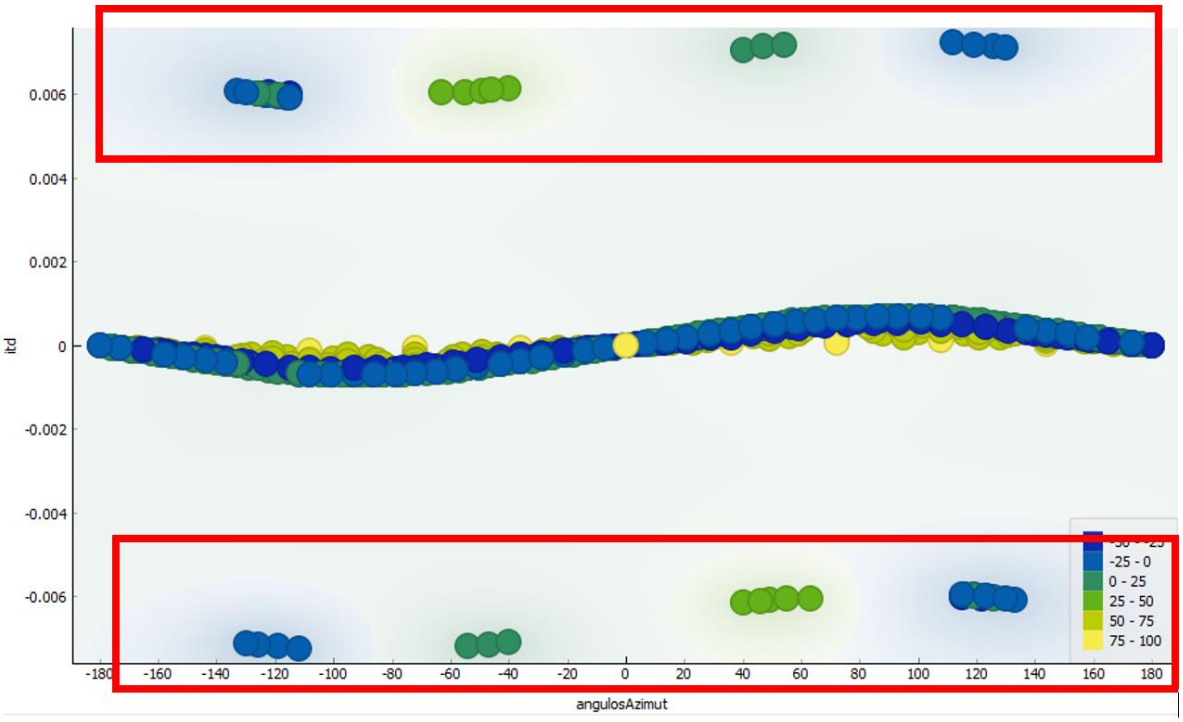


Nota. Gráfico de dispersión con valores del ITD vs angulosAzimut, con una diferencia de color según la característica angulosElevacion. Fuente: Realización propia (2021).

Lo que se puede notar es que entre el ITD y ambos ángulos no existe una relación. En este caso esto no debe ser correcto ya que el ITD es de las características que más información debe aportar acerca de la ubicación del sonido. Se puede notar además que existen datos atípicos como se ve en la figura 47. De este modo se extrae la correlación de Pearson tanto del ángulo de azimut y de elevación con las otras características.

Figura 47

Muestra de los valores atípicos



Nota. Muestra de los valores atípicos encontrados en los recuadros de color rojo. Fuente: Realización propia (2021).

Figura 48

Valores de la correlación

1	-0.757	angulosAzimut	diferenciaMaximosNivel
2	+0.736	angulosAzimut	ild
3	-0.699	amplitudIzquierda	angulosAzimut
4	+0.698	angulosAzimut	intensidadMinimalzquierda
5	-0.696	angulosAzimut	intensidadMaximalzquierda
6	+0.689	amplitudDerecha	angulosAzimut
7	-0.689	angulosAzimut	intensidadMinimaDerecha
8	+0.685	angulosAzimut	intensidadMaximaDerecha
9	+0.538	angulosAzimut	intensidadMedialzquierda
10	-0.537	angulosAzimut	intensidadMediaDerecha
11	+0.240	angulosAzimut	diferenciaMaximos
12	+0.093	angulosAzimut	itd
13	+0.028	angulosAzimut	intensidadMinima
14	-0.027	angulosAzimut	intensidadMaxima
15	+0.014	angulosAzimut	intensidadMedia
16	+0.009	angulosAzimut	angulosElevacion

Nota. Valores de la correlación del ángulo de azimut con las demás características. Fuente: Realización propia (2021).

Figura 49

Valores de la correlación

1	+0.696	angulosElevacion	intensidadMinima
2	-0.688	angulosElevacion	intensidadMaxima
3	+0.311	angulosElevacion	intensidadMedia
4	-0.299	angulosElevacion	intensidadMaximaDerecha
5	-0.299	angulosElevacion	intensidadMaximalzquierda
6	-0.288	amplitudIzquierda	angulosElevacion
7	-0.288	amplitudDerecha	angulosElevacion
8	+0.278	angulosElevacion	intensidadMinimalzquierda
9	+0.278	angulosElevacion	intensidadMinimaDerecha
10	+0.129	angulosElevacion	intensidadMedialzquierda
11	+0.129	angulosElevacion	intensidadMediaDerecha
12	+0.009	angulosAzimut	angulosElevacion
13	+0.000	angulosElevacion	diferenciaMaximos
14	-0.000	angulosElevacion	ild
15	-0.000	angulosElevacion	itd
16	+0.000	angulosElevacion	diferenciaMaximosNivel

Nota. Valores de la correlación del ángulo de elevación con las demás características.

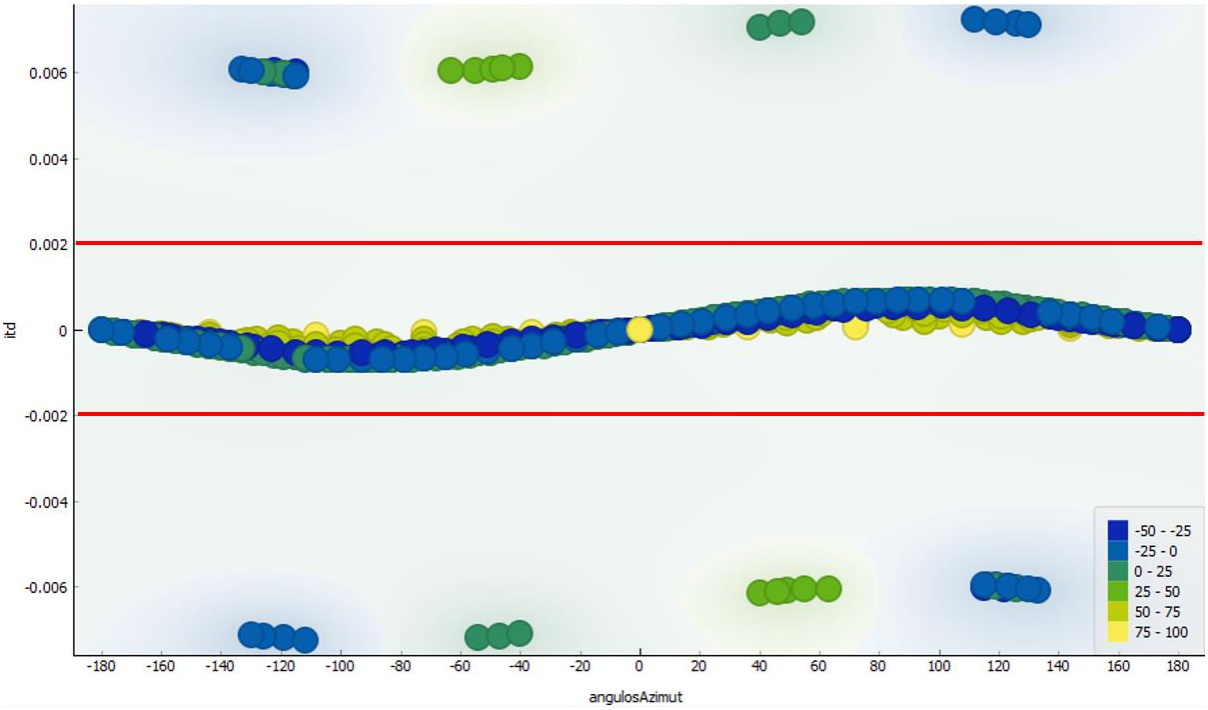
(Fuente: Realización propia (2021).

En el caso del ILD podemos ver que tiene una correlación bastante elevada con el ángulo de azimut. Esto tiene bastante sentido ya que esta característica debe aportar información relacionada a la dirección de procedencia del sonido. Mientras que el valor de la correlación del ITD básicamente demuestra que no existe ninguna relación con ninguno de los ángulos. Esto no

es coherente con la definición del ITD ya que este por lo menos debe tener correlación con uno de los dos ángulos. De este modo será necesario limpiar los datos y eliminar los registros que contienen los valores atípicos del ITD.

Figura 50

Muestra de los límites



Nota. Muestra de los límites representados con líneas rojas para dividir los valores atípicos del ITD.

Fuente: Realización propia (2021).

Una vez realizada la eliminación de los valores atípicos que son aquellos que sobrepasan el valor de 0,002000 y son menores que -0,002000 se puede revisar nuevamente si mejoró la correlación del ITD con al menos uno de los ángulos. En este

caso se eliminaron 48 registros, con un resultado final de 778 registros. Se puede observar que existe un cambio en la correlación entre el ITD con el ángulo de azimut (Figura 51).

Figura 51

Muestra de los límites

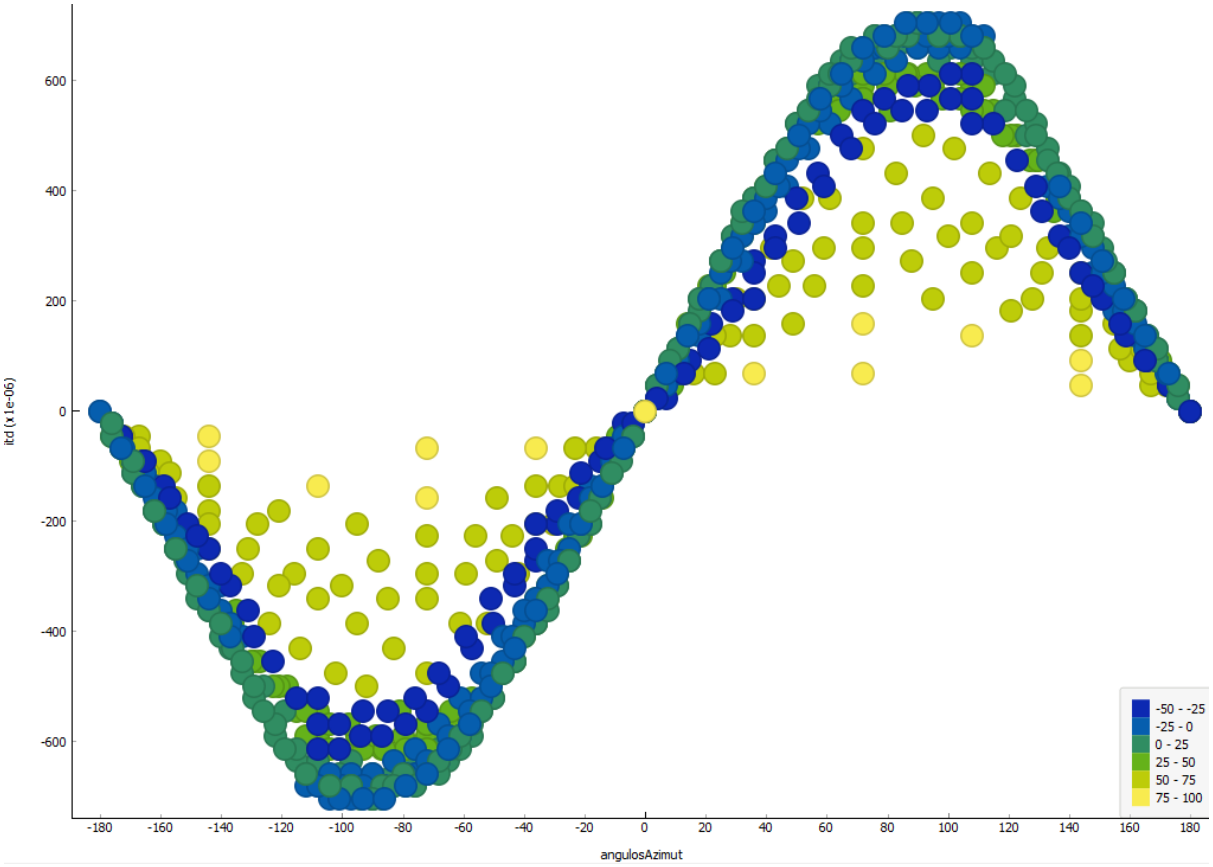
1	-0.742	angulosAzimut	diferenciaMaximosNivel
2	+0.728	angulosAzimut	itd
3	+0.720	angulosAzimut	ild
4	-0.683	amplitudIzquierda	angulosAzimut
5	+0.682	angulosAzimut	intensidadMinimalzquierda
6	-0.679	angulosAzimut	intensidadMaximalzquierda
7	+0.672	amplitudDerecha	angulosAzimut
8	-0.672	angulosAzimut	intensidadMinimaDerecha
9	+0.668	angulosAzimut	intensidadMaximaDerecha
10	+0.527	angulosAzimut	intensidadMedialzquierda
11	-0.526	angulosAzimut	intensidadMediaDerecha
12	+0.251	angulosAzimut	diferenciaMaximos
13	+0.028	angulosAzimut	intensidadMinima
14	-0.028	angulosAzimut	intensidadMaxima
15	+0.015	angulosAzimut	intensidadMedia
16	+0.009	angulosAzimut	angulosElevacion

Nota. Valores de la correlación del ángulo de azimut con las demás características.

Fuente: Realización propia (2021).

Figura 52

Gráfico de dispersión

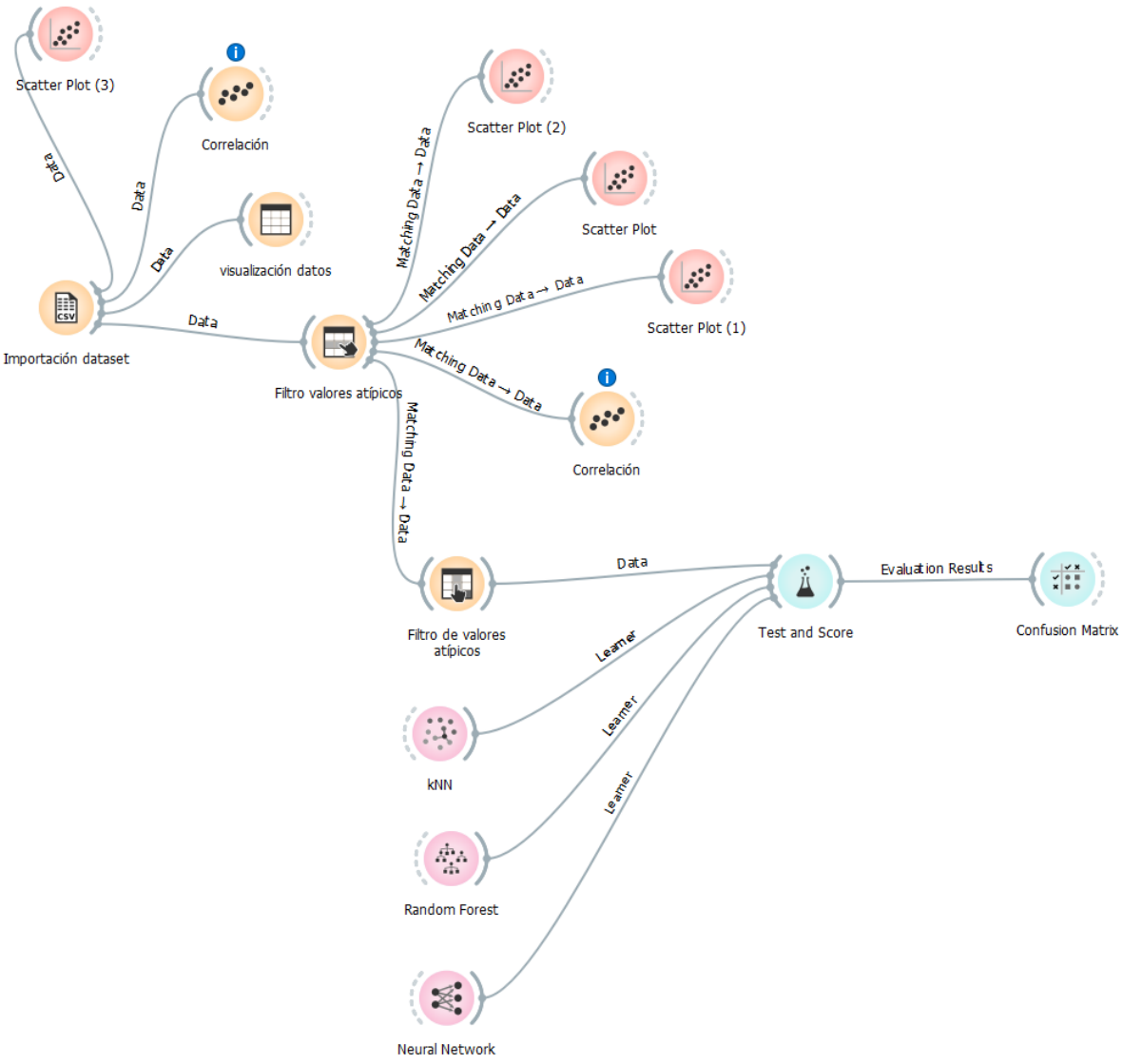


Nota. Gráfico de dispersión con valores del ITD vs angulosAzimut sin los valores atípicos, con una diferencia de color según la característica ángulos Elevación. Fuente: Realización propia (2021).

En este punto, una vez realizada la limpieza de los datos y revisado que no existan más valores atípicos se debe proceder al entrenamiento. En este caso la herramienta de Orange permite armar la arquitectura de todo este proceso mediante un flujo de manera gráfica. La arquitectura del modelo quedaría de la siguiente manera:

Figura 53

Arquitectura del entrenamiento del modelo en Orange



Fuente: Realización propia (2021).

4.2.ENTRENAMIENTO DEL MODELO

Para la parte del entrenamiento, se usa los tres modelos de los cuales se escogerá el que mejor precisión y resultado arroje. De igual manera durante esta etapa se entrena uno a uno los

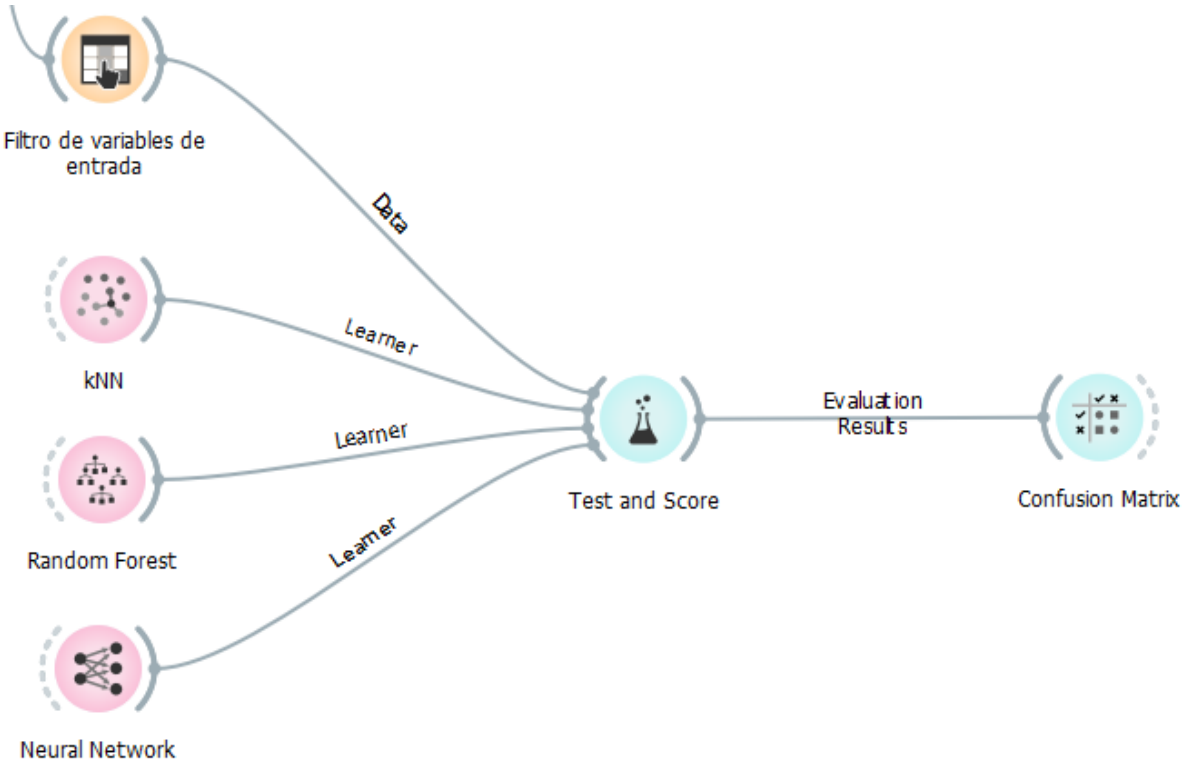
modelos y se modifican sus parámetros con el fin de conseguir siempre la mejor precisión posible. Para el dataset de entrenamiento se realizará una división porcentaje para tener un dataset de entrenamiento y otro dataset de validación. El dataset de entrenamiento será el 66% mientras que el dataset de validación será el 34%. Se lo realiza de esta manera para evitar el sobre entrenamiento u overfitting.

Para el tema del entrenamiento no todas las variables generadas se utilizarán para el entrenamiento, por lo que se solo se utiliza como inputs las siguientes variables:

1. diferenciaMaximosNivel
2. intensidadMinima
3. intensidadMaxima
4. amplitudIzquierda
5. intensidadMinimaIzquierda
6. intensidadMaximaIzquierda
7. amplitudDerecha
8. intensidadMinimaDerecha
9. intensidadMaximaDerecha
10. itd
11. ild
12. intensidadMediaDerecha
13. intensidadMediaIzquierda

Figura 54

Imagen flujo del entrenamiento con los tres modelos principales en la herramienta Orange



Fuente: Realización propia (2021).

4.3.PRUEBAS DE VALIDACIÓN DEL ENTRENAMIENTO

Una vez realizado el entrenamiento de cada uno de los modelos se describe por cada uno de ellos las diferentes modificaciones realizadas a sus parámetros y los diferentes valores de precisión que dio cada uno.

Modelo KN- Neighbors:

En el modelo KN-Neighbors se tiene como principal parámetro el número de vecinos cercanos con los que calcula la distancia con los demás registros. El segundo parámetro es el tipo de distancia entre las cuales está la distancia Euclidiana, Manhattan, entre otras. Sin embargo, para este problema se toma sólo estos dos tipos de distancias para entrenar el modelo. Los resultados se muestran en la tabla de la siguiente figura:

Figura 55

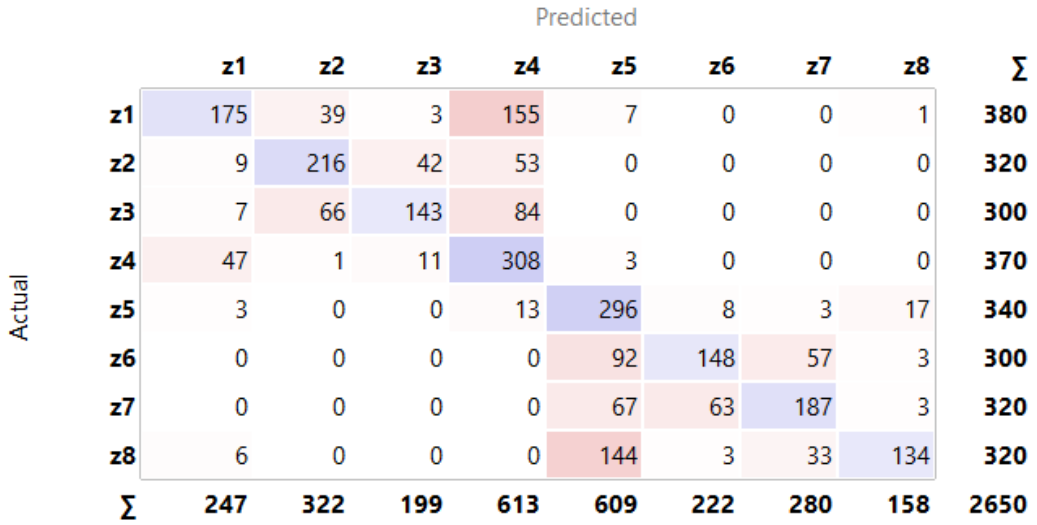
Resultados entrenamiento KNN

Distancia	Número vecinos	Precisión
Euclidiana	5	0,634
Euclidiana	7	0,638
Euclidiana	9	0,634
Euclidiana	11	0,633
Euclidiana	13	0,634
Euclidiana	15	0,628
Euclidiana	17	0,631
Euclidiana	19	0,639
Euclidiana	21	0,648
Euclidiana	23	0,639
Euclidiana	25	0,643
Euclidiana	27	0,651
Euclidiana	29	0,649
Euclidiana	31	0,647
Euclidiana	33	0,645
Euclidiana	35	0,65
Euclidiana	37	0,65
Euclidiana	39	0,644
Euclidiana	41	0,637
Euclidiana	43	0,634
Euclidiana	45	0,639
Euclidiana	47	0,636
Euclidiana	49	0,633
Manhattan	5	0,65
Manhattan	7	0,637
Manhattan	9	0,632
Manhattan	11	0,649
Manhattan	13	0,654
Manhattan	15	0,64
Manhattan	17	0,638
Manhattan	19	0,636
Manhattan	21	0,642
Manhattan	23	0,641
Manhattan	25	0,642
Manhattan	27	0,645
Manhattan	29	0,648
Manhattan	31	0,641
Manhattan	33	0,652
Manhattan	35	0,648
Manhattan	37	0,655
Manhattan	39	0,65
Manhattan	41	0,644
Manhattan	43	0,643
Manhattan	45	0,644
Manhattan	47	0,632
Manhattan	49	0,625

Fuente: Realización propia (2022).

Figura 56

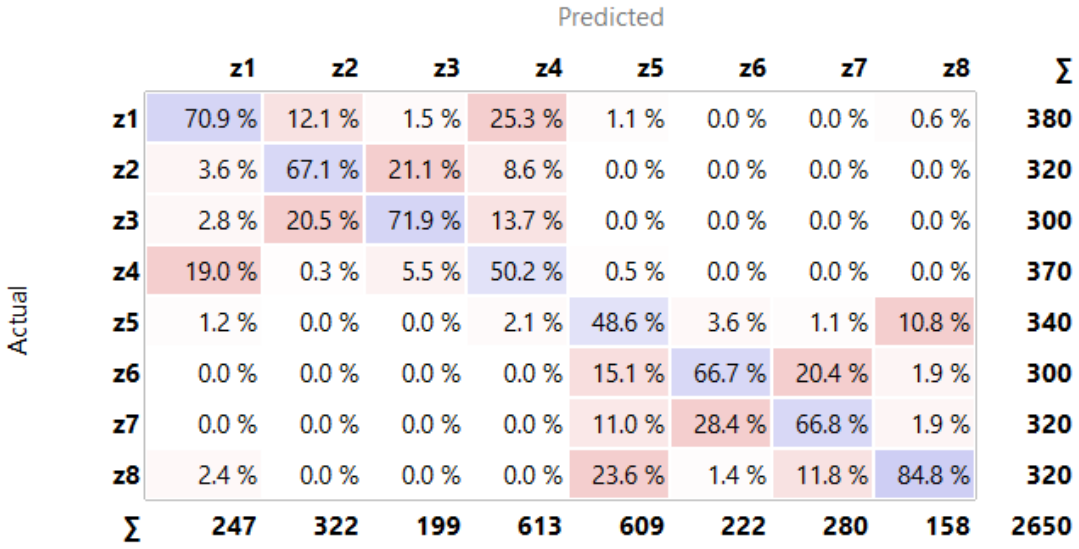
Matriz de confusión



Nota. Matriz de confusión resultado del entrenamiento KNN con distancia Manhattan y cinco vecinos cercanos. Fuente: Realización propia (2022).

Figura 57

Matriz de confusión en porcentajes



Nota. Matriz de confusión en porcentaje resultado del entrenamiento KNN con distancia Manhattan y cinco vecinos cercanos. (Fuente: Realización propia (2022).

Modelo Random Forest:

En el modelo Random Forest se tiene como principal parámetro el número árboles que se utiliza para entrenar el modelo. Para este caso lo que se hace es ir variando el valor del número de árboles y revisaremos con cuál de ellos se genera una precisión mayor. Con esto los resultados se muestran en la tabla de la siguiente figura:

Figura 58

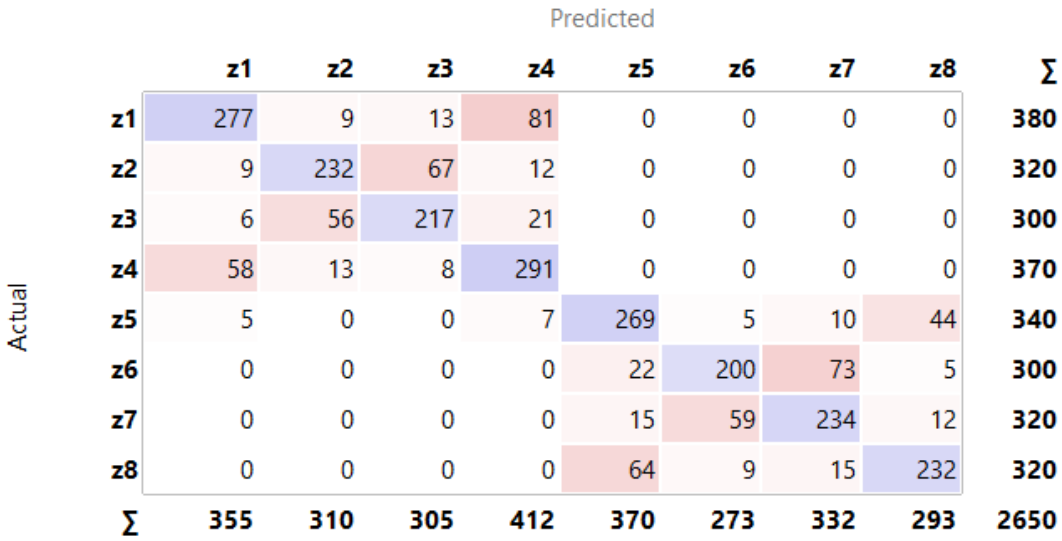
Resultados entrenamiento Random Forest

Árboles	Precisión
10	0,709
20	0,720
30	0,734
40	0,731
60	0,733
80	0,738
100	0,744
150	0,743
200	0,742
300	0,740
500	0,741
750	0,740
1000	0,741
2500	0,742
5000	0,743

Fuente: Realización propia (2022).

Figura 59

Matriz de confusión en porcentaje resultado del entrenamiento Random Forest con 100 árboles

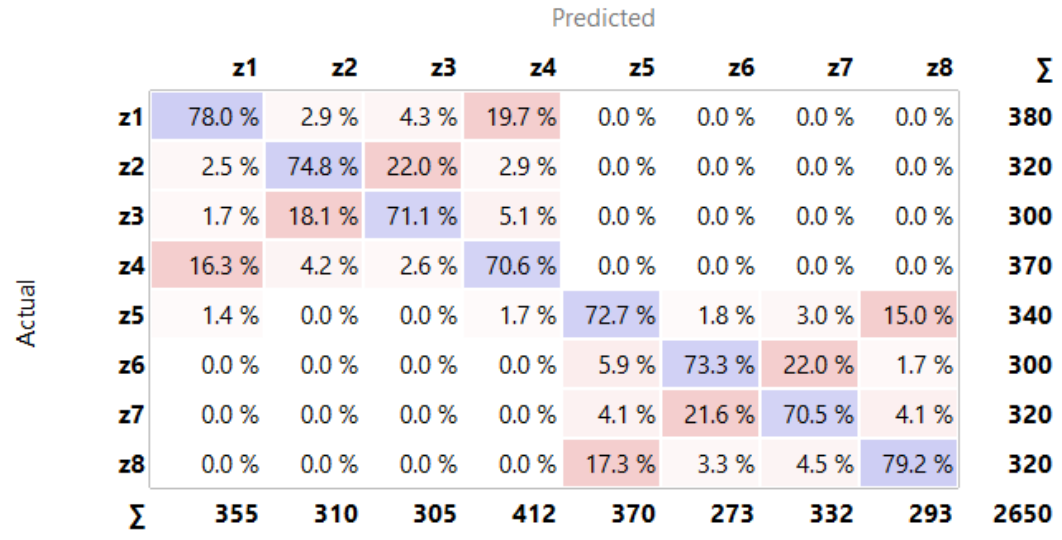


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	277	9	13	81	0	0	0	0	380
z2	9	232	67	12	0	0	0	0	320
z3	6	56	217	21	0	0	0	0	300
z4	58	13	8	291	0	0	0	0	370
z5	5	0	0	7	269	5	10	44	340
z6	0	0	0	0	22	200	73	5	300
z7	0	0	0	0	15	59	234	12	320
z8	0	0	0	0	64	9	15	232	320
∑	355	310	305	412	370	273	332	293	2650

Fuente: Realización propia (2022).

Figura 60

Matriz de confusión en porcentaje resultado del entrenamiento Random Forest con 100 árboles



Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	78.0 %	2.9 %	4.3 %	19.7 %	0.0 %	0.0 %	0.0 %	0.0 %	380
z2	2.5 %	74.8 %	22.0 %	2.9 %	0.0 %	0.0 %	0.0 %	0.0 %	320
z3	1.7 %	18.1 %	71.1 %	5.1 %	0.0 %	0.0 %	0.0 %	0.0 %	300
z4	16.3 %	4.2 %	2.6 %	70.6 %	0.0 %	0.0 %	0.0 %	0.0 %	370
z5	1.4 %	0.0 %	0.0 %	1.7 %	72.7 %	1.8 %	3.0 %	15.0 %	340
z6	0.0 %	0.0 %	0.0 %	0.0 %	5.9 %	73.3 %	22.0 %	1.7 %	300
z7	0.0 %	0.0 %	0.0 %	0.0 %	4.1 %	21.6 %	70.5 %	4.1 %	320
z8	0.0 %	0.0 %	0.0 %	0.0 %	17.3 %	3.3 %	4.5 %	79.2 %	320
∑	355	310	305	412	370	273	332	293	2650

Fuente: Realización propia (2022).

Modelo Neural Network:

En el modelo de redes neuronales tenemos como principal parámetro el número de neuronas y el número de capas. Además, tendremos otros parámetros que para este problema serán estáticos ya que se los define de forma óptima para lo que requiere este modelo. Para la función de activación dado que al ser un problema de clasificación se ha optado por la función de ReLu que es la función de unidad lineal rectificada, devuelve f(x) = max (0, x). El tipo de solucionador será 'adam' que se refiere a un optimizador estocástico basado en gradientes propuesto por Kingma, Diederik y Jimmy Ba. La tasa de aprendizaje inicial utilizada será de 0.001 la cual controla el tamaño del paso en la actualización de los pesos. Con esto los resultados se muestran en la tabla de la siguiente figura:

Figura 61

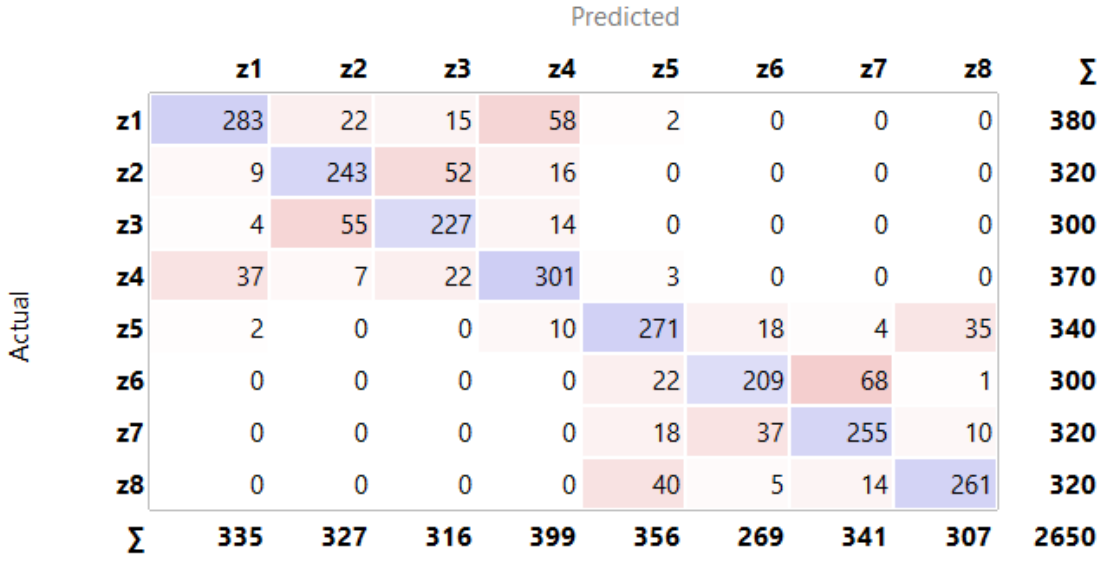
Resultados entrenamiento Red Neuronal

Número de capas	Número de neuronas por capa	Precisión
1	50	0,743
1	100	0,762
1	250	0,772
1	500	0,777
1	1000	0,775
1	2000	0,776
2	50	0,766
2	100	0,771
2	250	0,774
2	500	0,768
2	1000	0,765
2	2000	0,766
3	50	0,752
3	100	0,771
3	250	0,765
3	500	0,769
3	1000	0,776
3	2000	0,766
4	50	0,766
4	100	0,769
4	250	0,762
4	500	0,760
4	1000	0,767
4	2000	0,764
5	50	0,736
5	100	0,754
5	250	0,755
5	500	0,765
5	1000	0,752

Fuente: Realización propia (2022).

Figura 62

Matriz de confusión resultado del entrenamiento de Red Neuronal con dos capas de 2000 neuronas

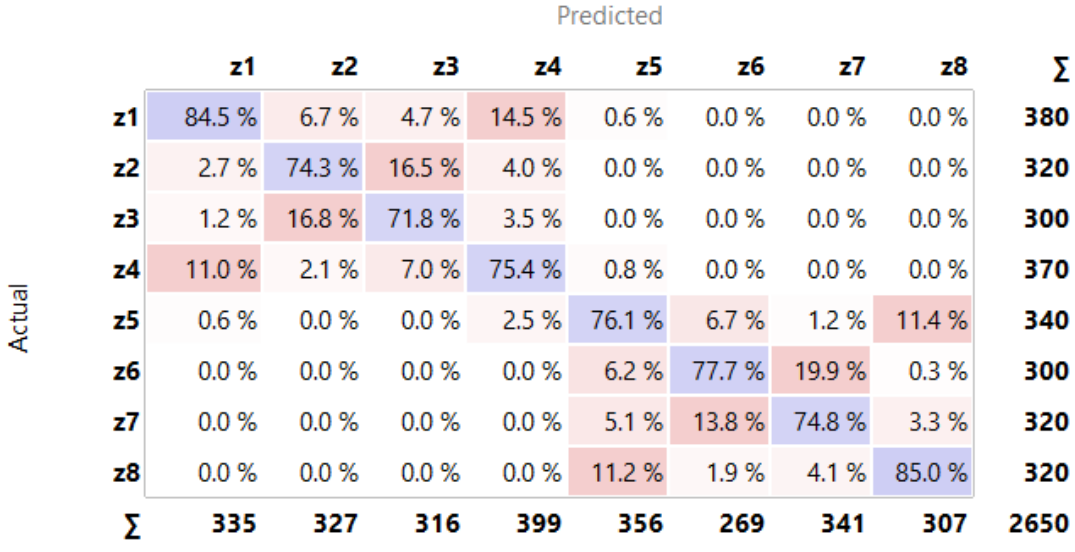


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	283	22	15	58	2	0	0	0	380
z2	9	243	52	16	0	0	0	0	320
z3	4	55	227	14	0	0	0	0	300
z4	37	7	22	301	3	0	0	0	370
z5	2	0	0	10	271	18	4	35	340
z6	0	0	0	0	22	209	68	1	300
z7	0	0	0	0	18	37	255	10	320
z8	0	0	0	0	40	5	14	261	320
∑	335	327	316	399	356	269	341	307	2650

Fuente: Realización propia (2022).

Figura 63

Matriz de confusión en porcentaje resultado del entrenamiento de Red Neuronal con dos capas de 2000 neuronas



Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	84.5 %	6.7 %	4.7 %	14.5 %	0.6 %	0.0 %	0.0 %	0.0 %	380
z2	2.7 %	74.3 %	16.5 %	4.0 %	0.0 %	0.0 %	0.0 %	0.0 %	320
z3	1.2 %	16.8 %	71.8 %	3.5 %	0.0 %	0.0 %	0.0 %	0.0 %	300
z4	11.0 %	2.1 %	7.0 %	75.4 %	0.8 %	0.0 %	0.0 %	0.0 %	370
z5	0.6 %	0.0 %	0.0 %	2.5 %	76.1 %	6.7 %	1.2 %	11.4 %	340
z6	0.0 %	0.0 %	0.0 %	0.0 %	6.2 %	77.7 %	19.9 %	0.3 %	300
z7	0.0 %	0.0 %	0.0 %	0.0 %	5.1 %	13.8 %	74.8 %	3.3 %	320
z8	0.0 %	0.0 %	0.0 %	0.0 %	11.2 %	1.9 %	4.1 %	85.0 %	320
∑	335	327	316	399	356	269	341	307	2650

Fuente: Realización propia (2022).

Los resultados de la máxima precisión alcanzada por cada modelo son las siguientes:

- KN- Neighbors: 65,5%
- Random Forest: 74,4%
- Neural Network: 77,7%

Se observa que los modelos más precisos son el de Random Forest y el de Neural Network. Sin embargo, a pesar de que el modelo de KN-Neighbors no consiguió la precisión solicitada en los requerimientos se procederá a ocupar este modelo en las pruebas de validación ya que su precisión se acerca al 70%.

5. RESULTADOS

5.1. PRUEBAS DE VALIDACIÓN DEL MODELO

Para las pruebas de validación del modelo se hace uso de cuatro audios diferentes al audio utilizado en el entrenamiento. A partir de estos audios se prepara un dataset por cada uno de la misma forma como se lo hizo con el audio utilizado en el entrenamiento. Se generaron 826 audios por cada uno con diferentes direcciones de procedencia uniformemente distribuidos.

Para estas pruebas se extraerá la precisión de los modelos de tres formas:

1. Precisión: El porcentaje de la división entre el número de registros que el modelo reconoció la zona correctamente y el total de registros.
2. Precisión con errores tipo A: Los errores tipo A son aquellos que corresponden al problema del cono de confusión (Figura 1). En este caso no se tomará en cuenta los casos en el que el modelo se confunda entre las siguientes zonas: z1-z4, z2-z3, z5-z8 y z6-z7. Esta precisión será la suma de la precisión más la precisión sin tomar en cuenta los errores tipo A.
3. Precisión con errores tipo B: Los errores tipo B son aquellos el modelo no reconoce correctamente la zona y lo identifica como la zona adyacente. En este caso no se tomará en cuenta los casos en el que el modelo se confunda entre las siguientes zonas: z1-z2, z3-z4, z5-z6, z7-z8. Esta precisión será la suma de la precisión con errores tipo A más la precisión sin tomar en cuenta los errores tipo B.

Los errores que no entren en la categoría de tipo A o B se consideran inaceptables. De esta forma se obtuvo los siguientes valores al realizar las pruebas con los modelos y con los audios de prueba:

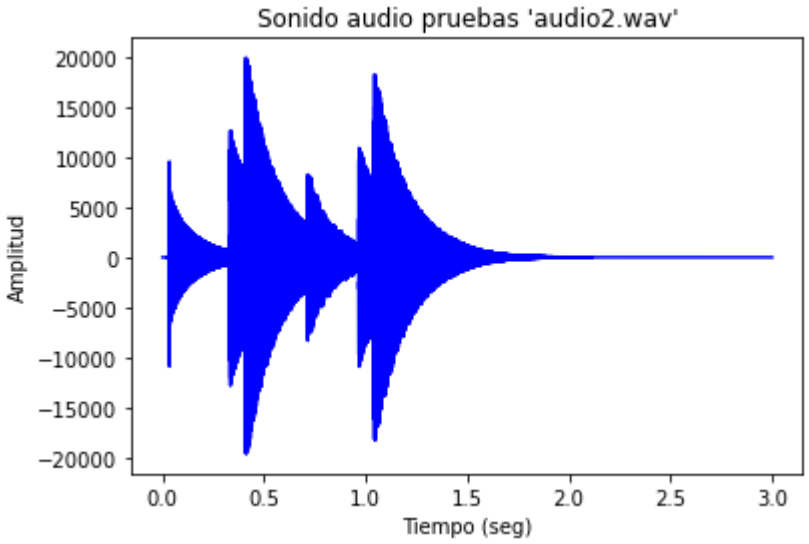
Prueba 1:

Nombre del audio: audio2.wav

Onda del sonido del audio de prueba:

Figura 64

Gráfico de la onda del sonido de prueba 'audio2.wav'



Fuente: Realización propia (2022).

Resultado de las pruebas:

Figura 65

Tabla de resultado de las pruebas

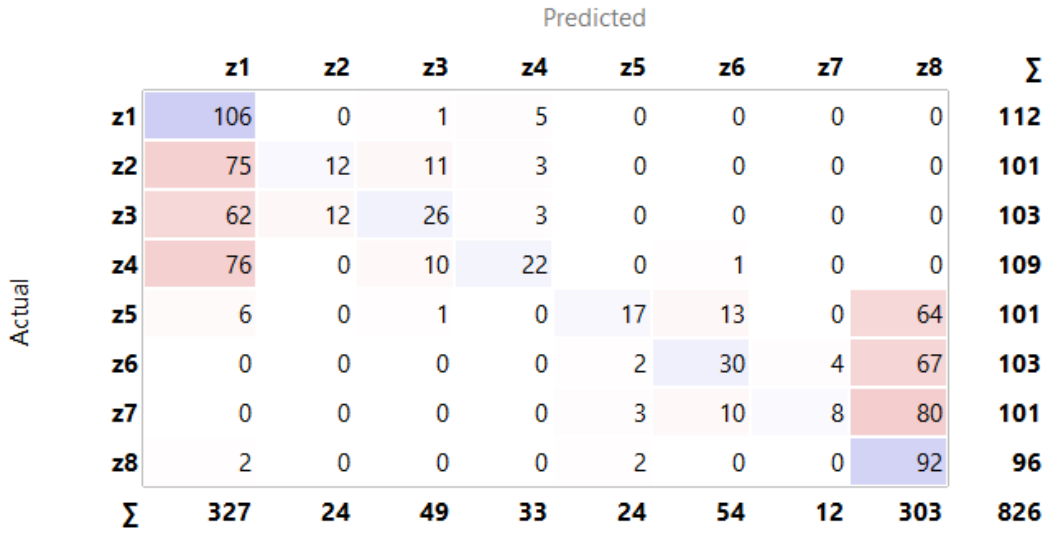
Modelo	Precisión	P tipo A	P tipo B
Neural Network	0,532	0,772	0,975
Random Forest	0,502	0,846	0,926
KNN	0,632	0,873	0,944

Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo Neural Network:

Figura 66

Matriz de confusión resultado de las pruebas con Red Neuronal

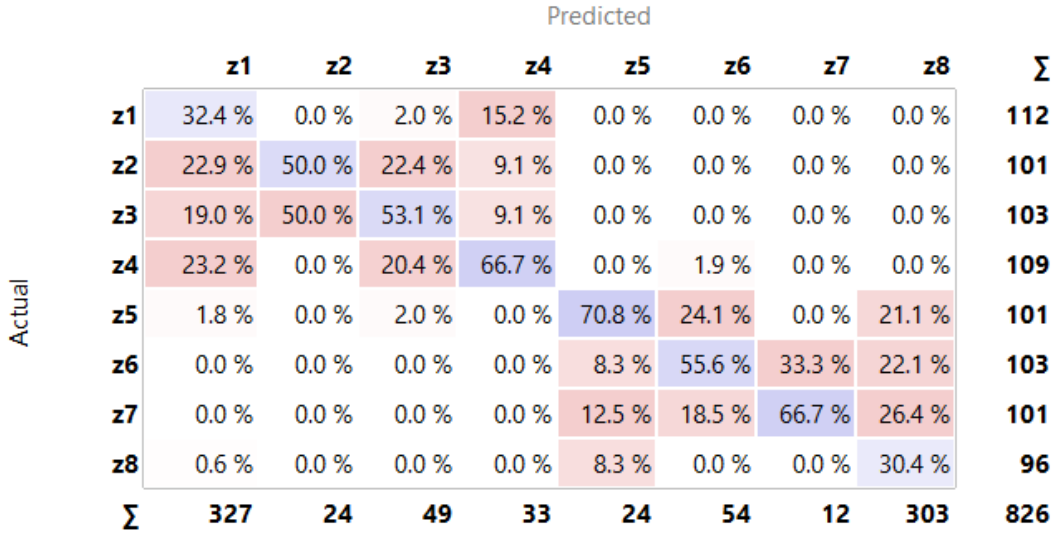


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	106	0	1	5	0	0	0	0	112
z2	75	12	11	3	0	0	0	0	101
z3	62	12	26	3	0	0	0	0	103
z4	76	0	10	22	0	1	0	0	109
z5	6	0	1	0	17	13	0	64	101
z6	0	0	0	0	2	30	4	67	103
z7	0	0	0	0	3	10	8	80	101
z8	2	0	0	0	2	0	0	92	96
∑	327	24	49	33	24	54	12	303	826

Fuente: Realización propia (2022).

Figura 67

Matriz de confusión en porcentaje resultado de las pruebas con Red Neuronal

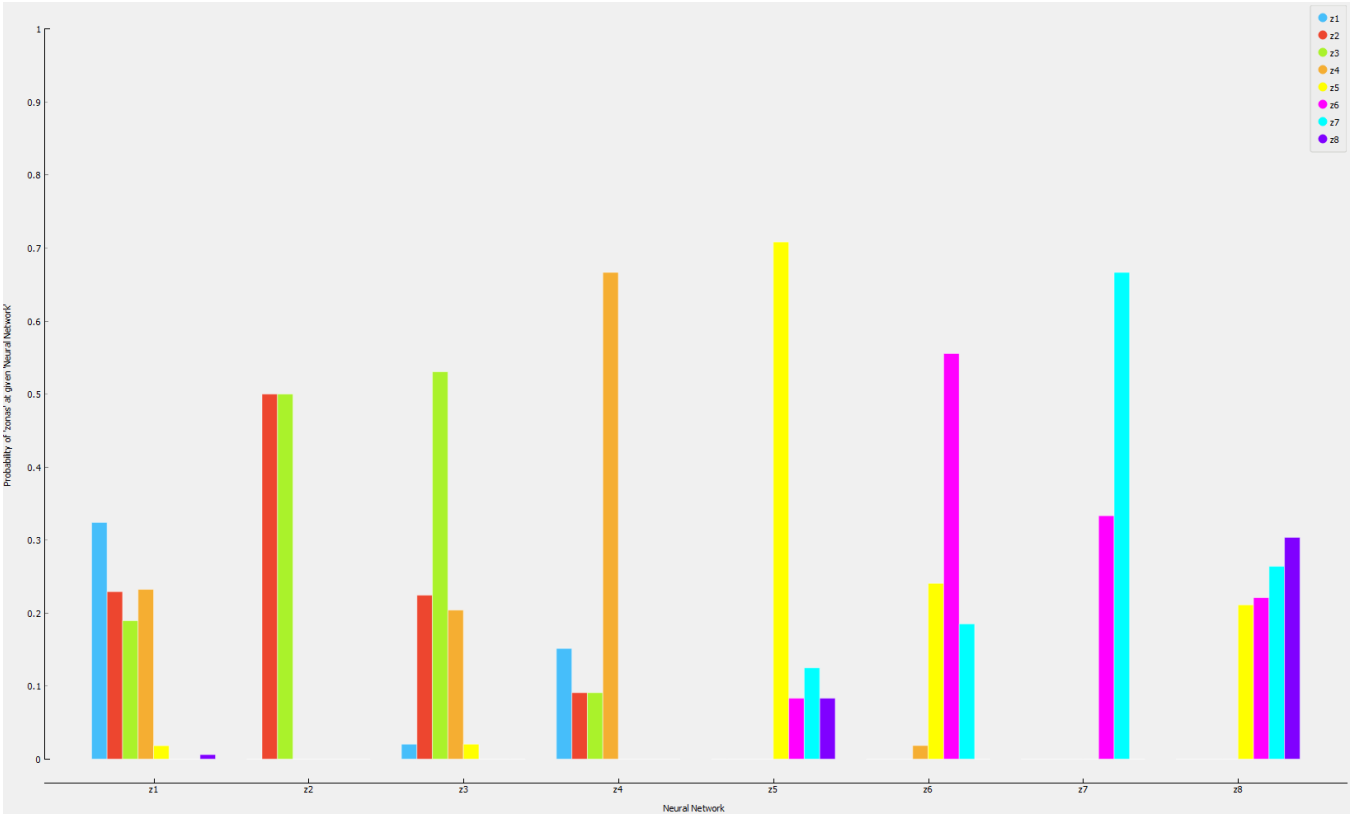


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	32.4 %	0.0 %	2.0 %	15.2 %	0.0 %	0.0 %	0.0 %	0.0 %	112
z2	22.9 %	50.0 %	22.4 %	9.1 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	19.0 %	50.0 %	53.1 %	9.1 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	23.2 %	0.0 %	20.4 %	66.7 %	0.0 %	1.9 %	0.0 %	0.0 %	109
z5	1.8 %	0.0 %	2.0 %	0.0 %	70.8 %	24.1 %	0.0 %	21.1 %	101
z6	0.0 %	0.0 %	0.0 %	0.0 %	8.3 %	55.6 %	33.3 %	22.1 %	103
z7	0.0 %	0.0 %	0.0 %	0.0 %	12.5 %	18.5 %	66.7 %	26.4 %	101
z8	0.6 %	0.0 %	0.0 %	0.0 %	8.3 %	0.0 %	0.0 %	30.4 %	96
∑	327	24	49	33	24	54	12	303	826

Fuente: Realización propia (2022).

Figura 68

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Red Neuronal

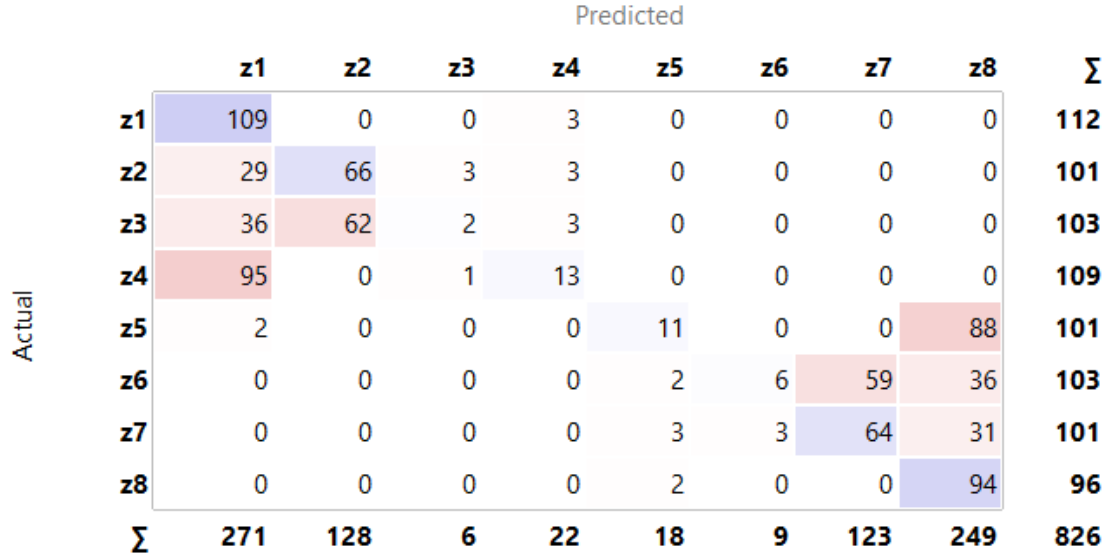


Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo Random Forest:

Figura 69

Matriz de confusión resultado de las pruebas con Random Forest

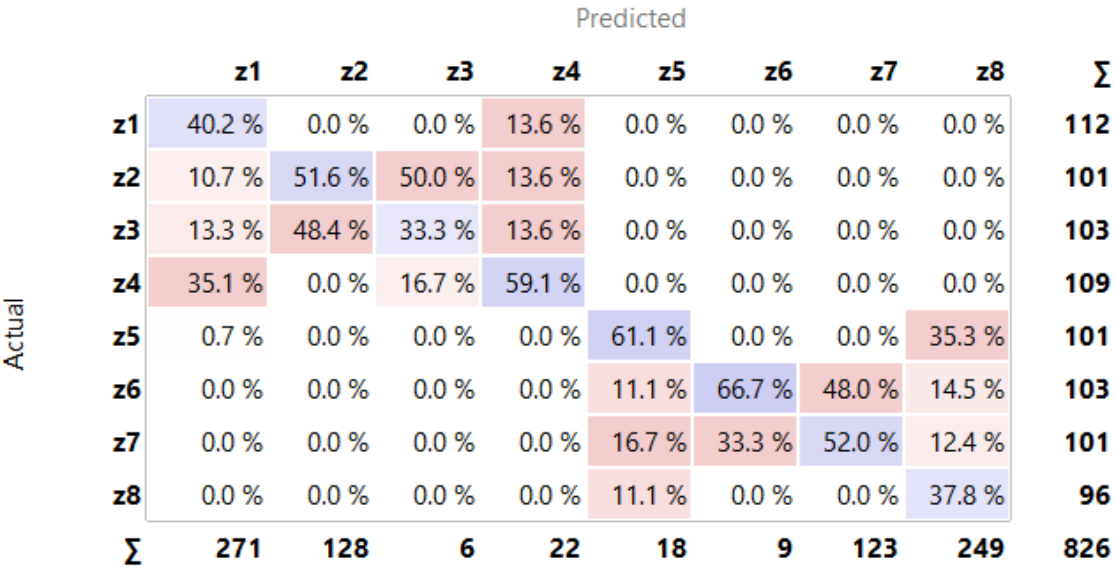


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	109	0	0	3	0	0	0	0	112
z2	29	66	3	3	0	0	0	0	101
z3	36	62	2	3	0	0	0	0	103
z4	95	0	1	13	0	0	0	0	109
z5	2	0	0	0	11	0	0	88	101
z6	0	0	0	0	2	6	59	36	103
z7	0	0	0	0	3	3	64	31	101
z8	0	0	0	0	2	0	0	94	96
∑	271	128	6	22	18	9	123	249	826

Fuente: Realización propia (2022).

Figura 70

Matriz de confusión en porcentaje resultado de las pruebas con Random Forest

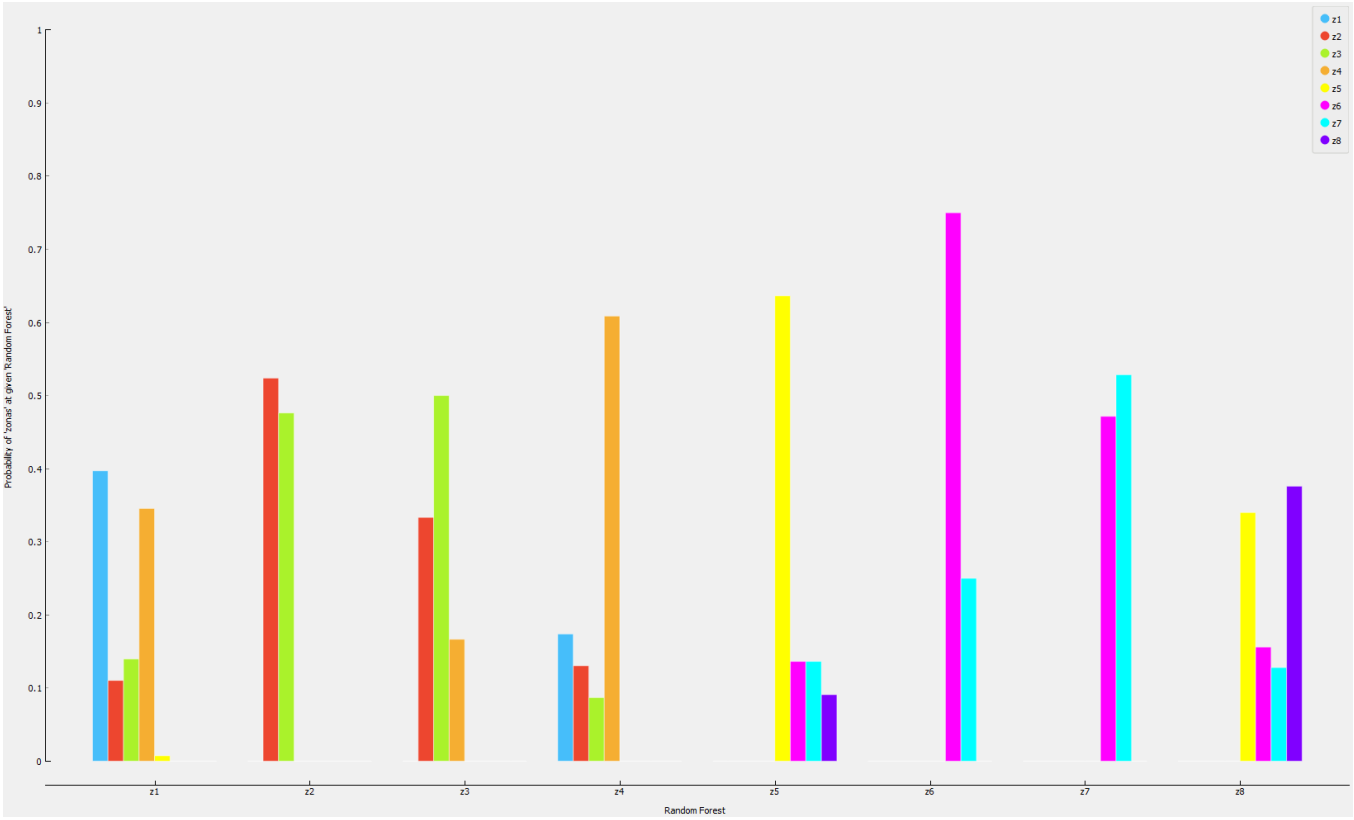


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	40.2 %	0.0 %	0.0 %	13.6 %	0.0 %	0.0 %	0.0 %	0.0 %	112
z2	10.7 %	51.6 %	50.0 %	13.6 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	13.3 %	48.4 %	33.3 %	13.6 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	35.1 %	0.0 %	16.7 %	59.1 %	0.0 %	0.0 %	0.0 %	0.0 %	109
z5	0.7 %	0.0 %	0.0 %	0.0 %	61.1 %	0.0 %	0.0 %	35.3 %	101
z6	0.0 %	0.0 %	0.0 %	0.0 %	11.1 %	66.7 %	48.0 %	14.5 %	103
z7	0.0 %	0.0 %	0.0 %	0.0 %	16.7 %	33.3 %	52.0 %	12.4 %	101
z8	0.0 %	0.0 %	0.0 %	0.0 %	11.1 %	0.0 %	0.0 %	37.8 %	96
∑	271	128	6	22	18	9	123	249	826

Fuente: Realización propia (2022).

Figura 71

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Random Forest

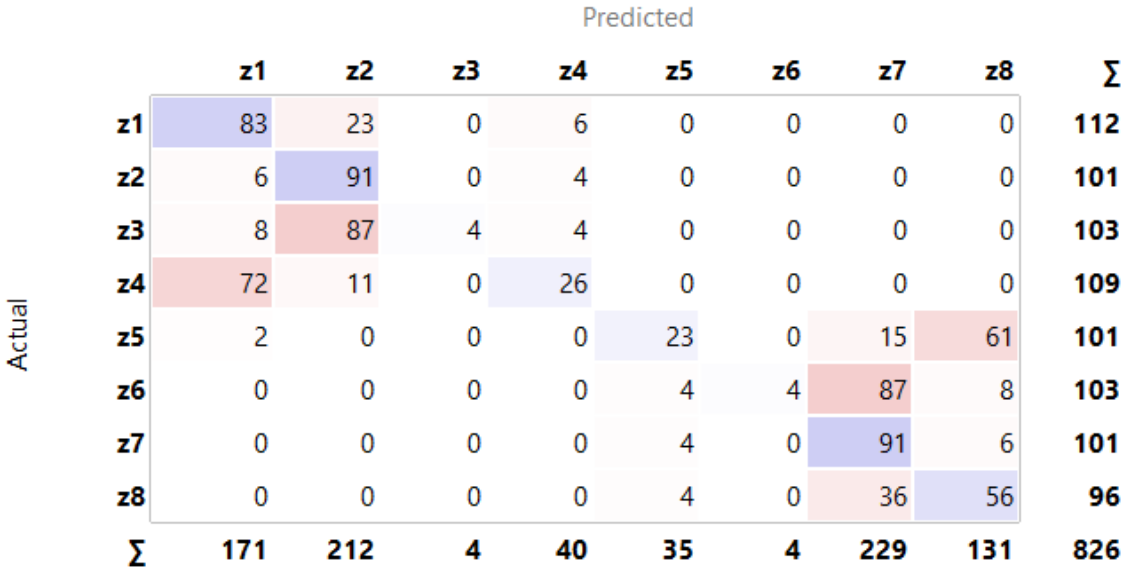


Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo KNN:

Figura 72

Matriz de confusión resultado de las pruebas con KNN

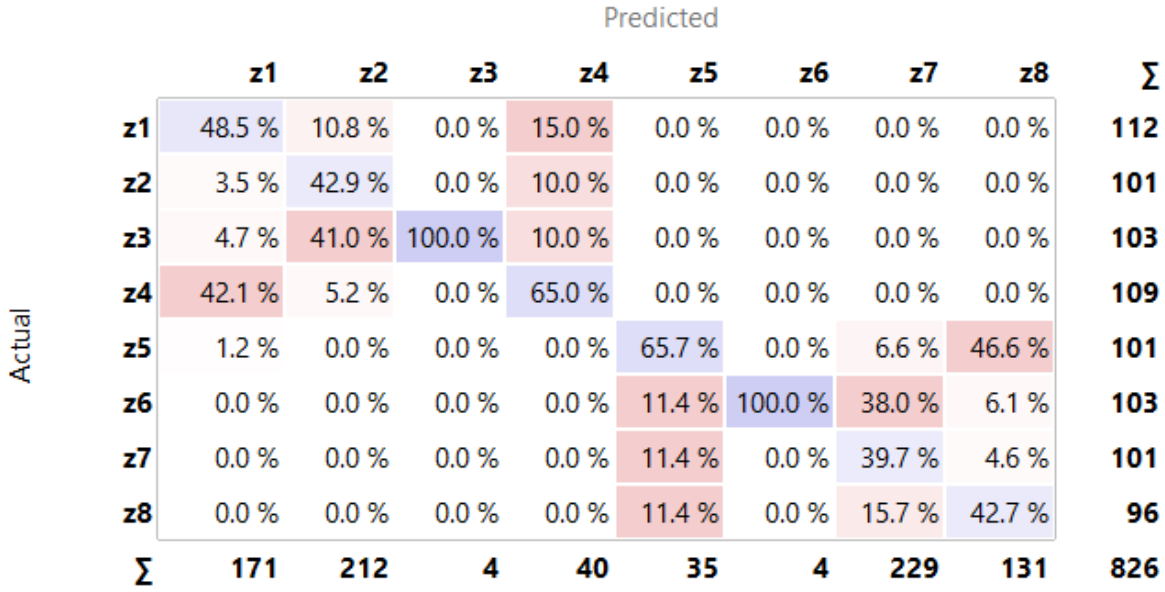


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	83	23	0	6	0	0	0	0	112
z2	6	91	0	4	0	0	0	0	101
z3	8	87	4	4	0	0	0	0	103
z4	72	11	0	26	0	0	0	0	109
z5	2	0	0	0	23	0	15	61	101
z6	0	0	0	0	4	4	87	8	103
z7	0	0	0	0	4	0	91	6	101
z8	0	0	0	0	4	0	36	56	96
∑	171	212	4	40	35	4	229	131	826

Fuente: Realización propia (2022).

Figura 73

Matriz de confusión en porcentaje resultado de las pruebas con KNN

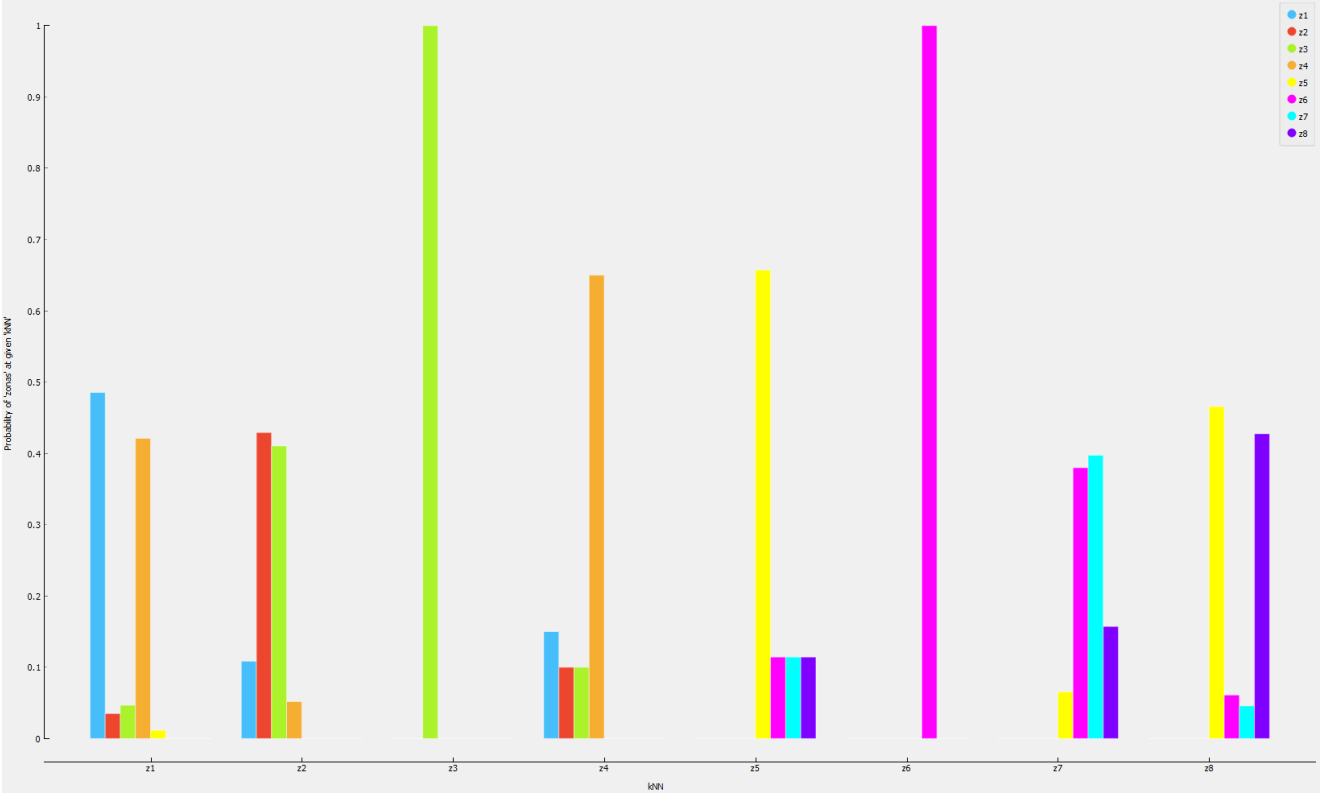


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	48.5 %	10.8 %	0.0 %	15.0 %	0.0 %	0.0 %	0.0 %	0.0 %	112
z2	3.5 %	42.9 %	0.0 %	10.0 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	4.7 %	41.0 %	100.0 %	10.0 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	42.1 %	5.2 %	0.0 %	65.0 %	0.0 %	0.0 %	0.0 %	0.0 %	109
z5	1.2 %	0.0 %	0.0 %	0.0 %	65.7 %	0.0 %	6.6 %	46.6 %	101
z6	0.0 %	0.0 %	0.0 %	0.0 %	11.4 %	100.0 %	38.0 %	6.1 %	103
z7	0.0 %	0.0 %	0.0 %	0.0 %	11.4 %	0.0 %	39.7 %	4.6 %	101
z8	0.0 %	0.0 %	0.0 %	0.0 %	11.4 %	0.0 %	15.7 %	42.7 %	96
∑	171	212	4	40	35	4	229	131	826

Fuente: Realización propia (2022).

Figura 74

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de KNN



Fuente: Realización propia (2022).

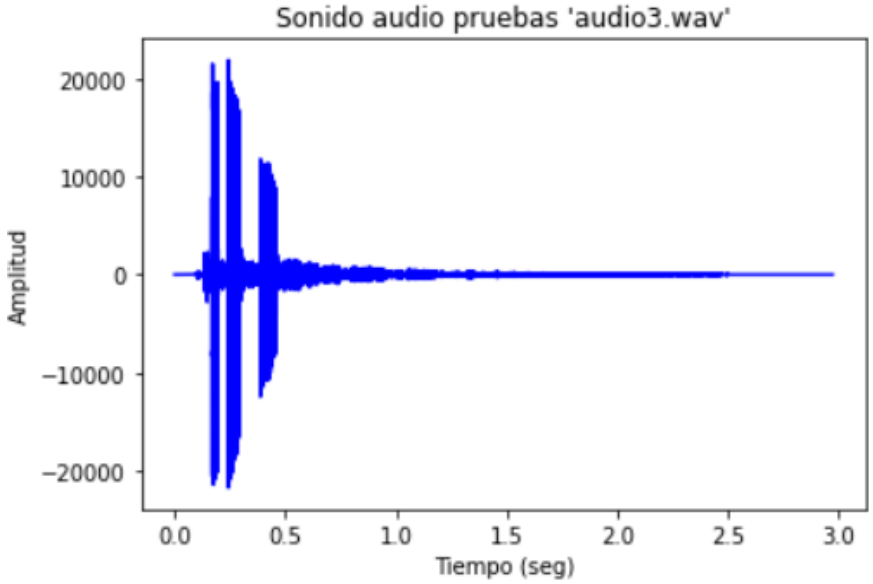
Prueba 2:

Nombre del audio: audio3.wav

Onda del sonido del audio de prueba:

Figura 75

Gráfico de la onda del sonido de prueba 'audio3.wav'



Fuente: Realización propia (2022).

Resultado de las pruebas:

Figura 76

Tabla de resultado de las pruebas

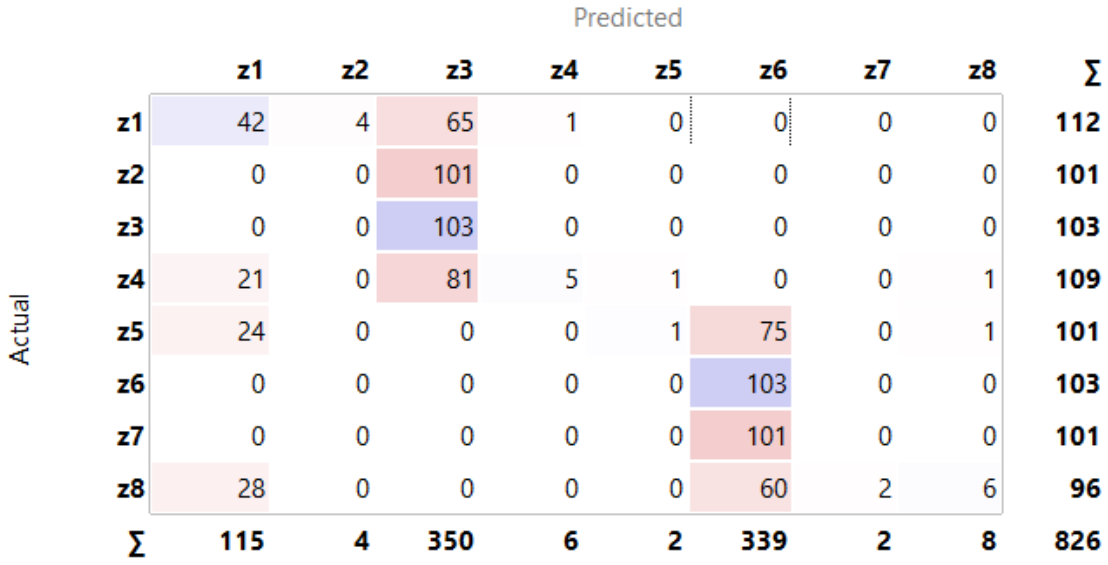
Modelo	Precisión	P tipo A	P tipo B
Neural Network	0,382	0,514	0,82
Random Forest	0,497	0,733	0,898
KNN	0,604	0,812	0,882

Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo Neural Network:

Figura 77

Matriz de confusión resultado de las pruebas con Red Neuronal

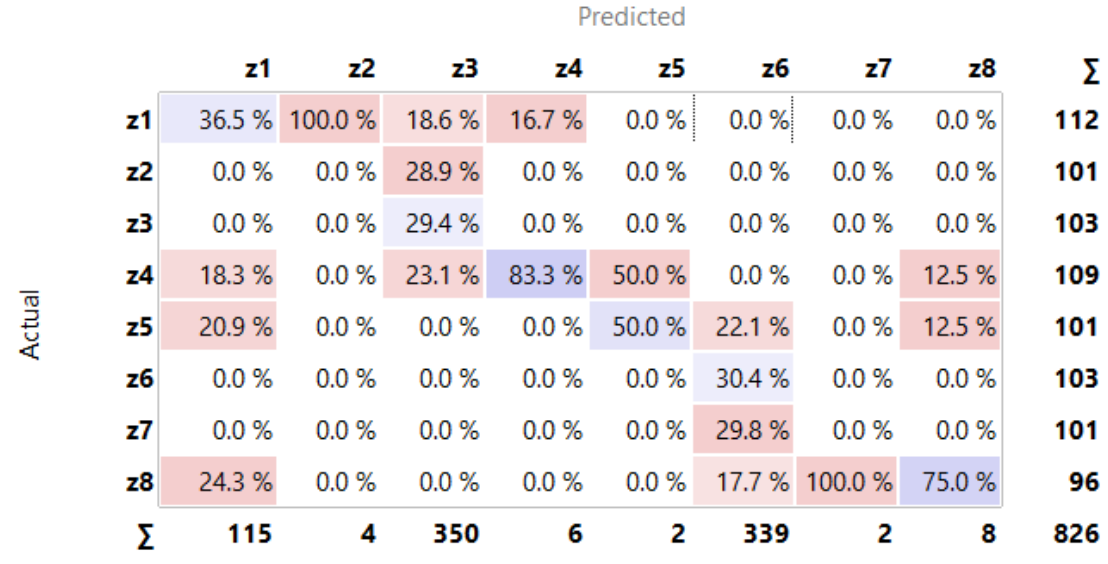


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	42	4	65	1	0	0	0	0	112
z2	0	0	101	0	0	0	0	0	101
z3	0	0	103	0	0	0	0	0	103
z4	21	0	81	5	1	0	0	1	109
z5	24	0	0	0	1	75	0	1	101
z6	0	0	0	0	0	103	0	0	103
z7	0	0	0	0	0	101	0	0	101
z8	28	0	0	0	0	60	2	6	96
∑	115	4	350	6	2	339	2	8	826

Fuente: Realización propia (2022).

Figura 78

Matriz de confusión en porcentaje resultado de las pruebas con Red Neuronal

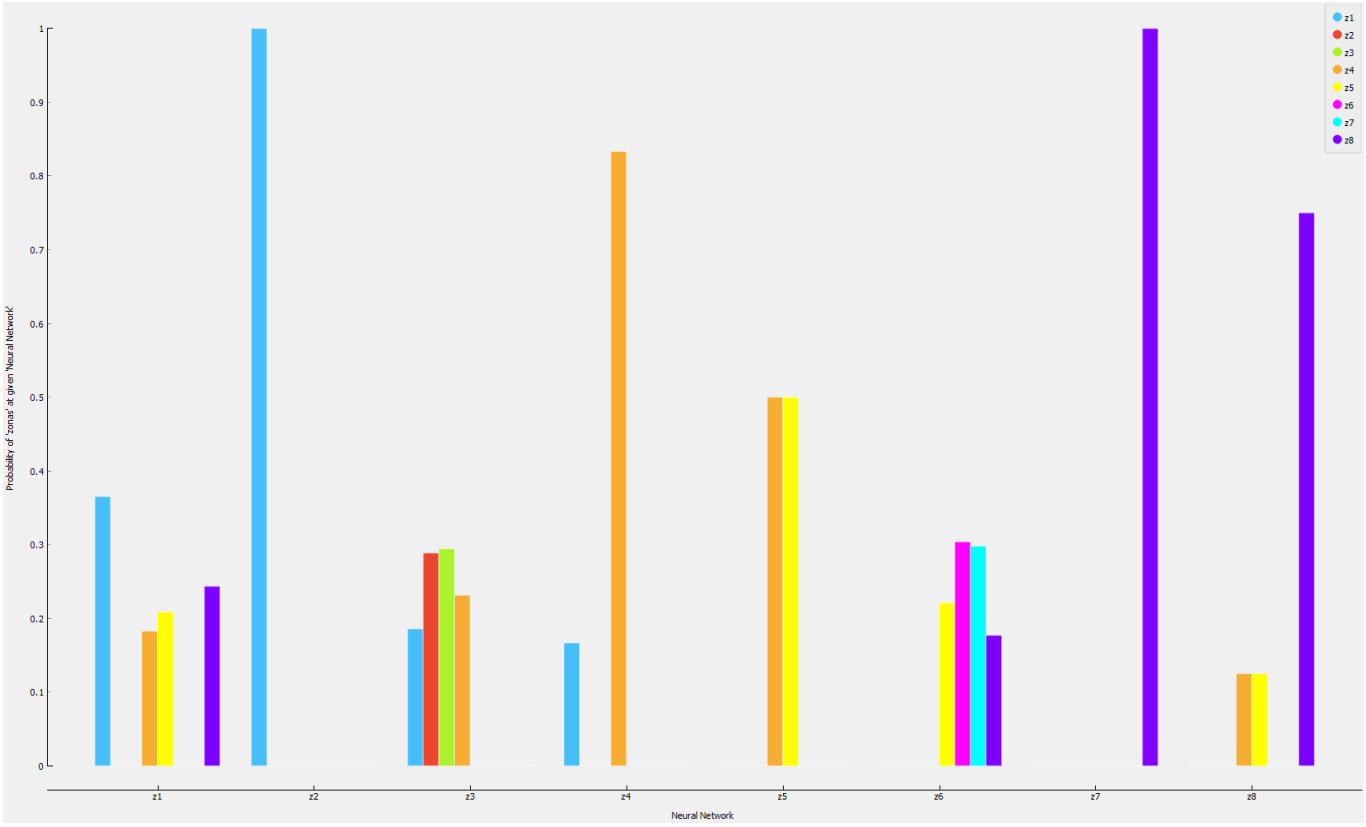


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	36.5 %	100.0 %	18.6 %	16.7 %	0.0 %	0.0 %	0.0 %	0.0 %	112
z2	0.0 %	0.0 %	28.9 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	0.0 %	0.0 %	29.4 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	18.3 %	0.0 %	23.1 %	83.3 %	50.0 %	0.0 %	0.0 %	12.5 %	109
z5	20.9 %	0.0 %	0.0 %	0.0 %	50.0 %	22.1 %	0.0 %	12.5 %	101
z6	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	30.4 %	0.0 %	0.0 %	103
z7	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	29.8 %	0.0 %	0.0 %	101
z8	24.3 %	0.0 %	0.0 %	0.0 %	0.0 %	17.7 %	100.0 %	75.0 %	96
∑	115	4	350	6	2	339	2	8	826

Fuente: Realización propia (2022).

Figura 79

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Red Neuronal

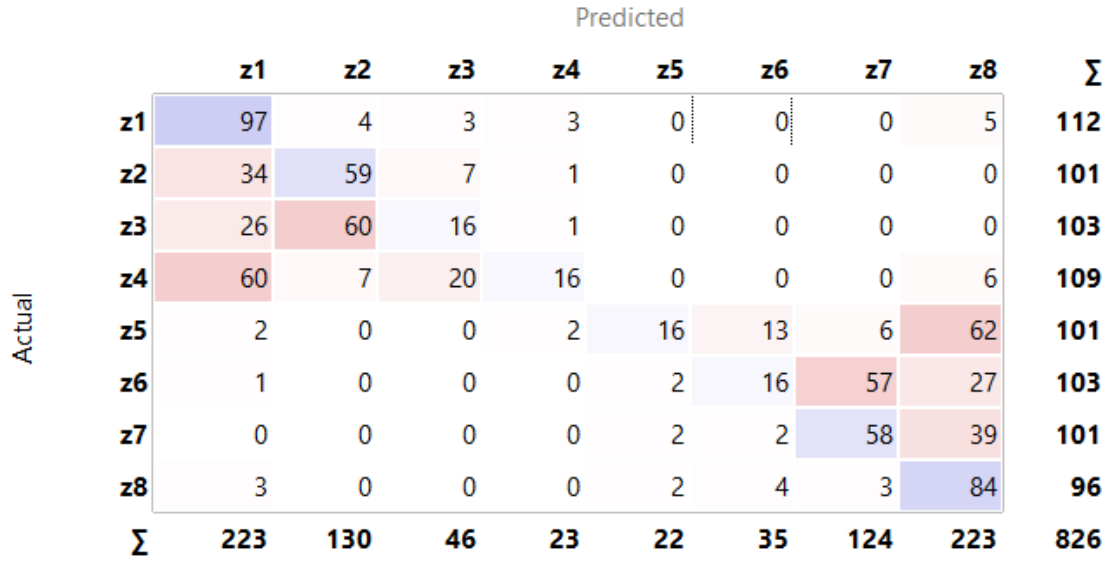


Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo Random Forest:

Figura 80

Matriz de confusión resultado de las pruebas con Random Forest

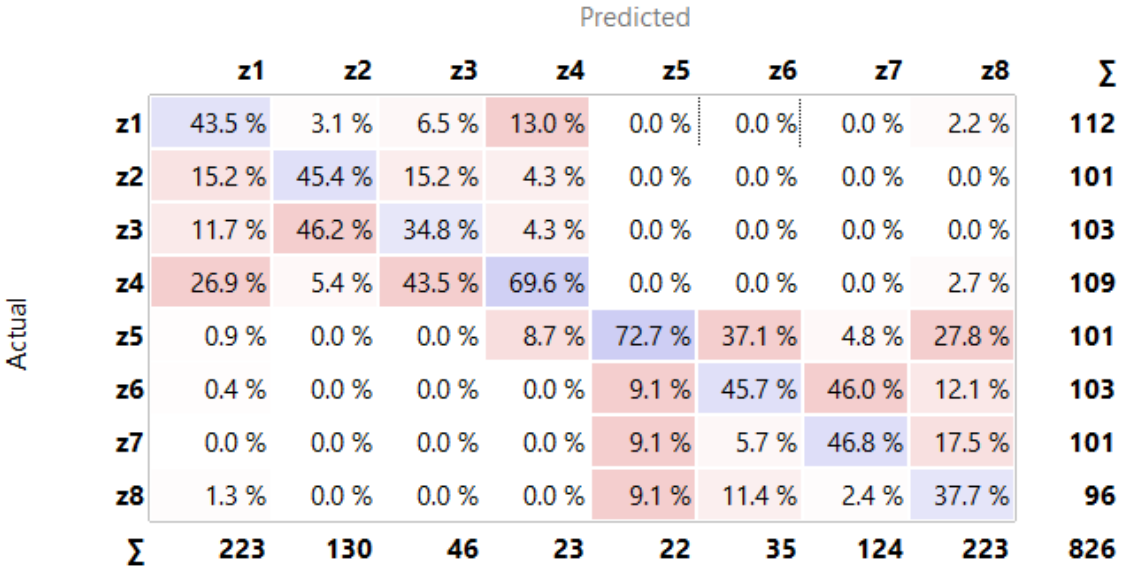


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	97	4	3	3	0	0	0	5	112
z2	34	59	7	1	0	0	0	0	101
z3	26	60	16	1	0	0	0	0	103
z4	60	7	20	16	0	0	0	6	109
z5	2	0	0	2	16	13	6	62	101
z6	1	0	0	0	2	16	57	27	103
z7	0	0	0	0	2	2	58	39	101
z8	3	0	0	0	2	4	3	84	96
∑	223	130	46	23	22	35	124	223	826

Fuente: Realización propia (2022).

Figura 81

Matriz de confusión en porcentaje resultado de las pruebas con Random Forest

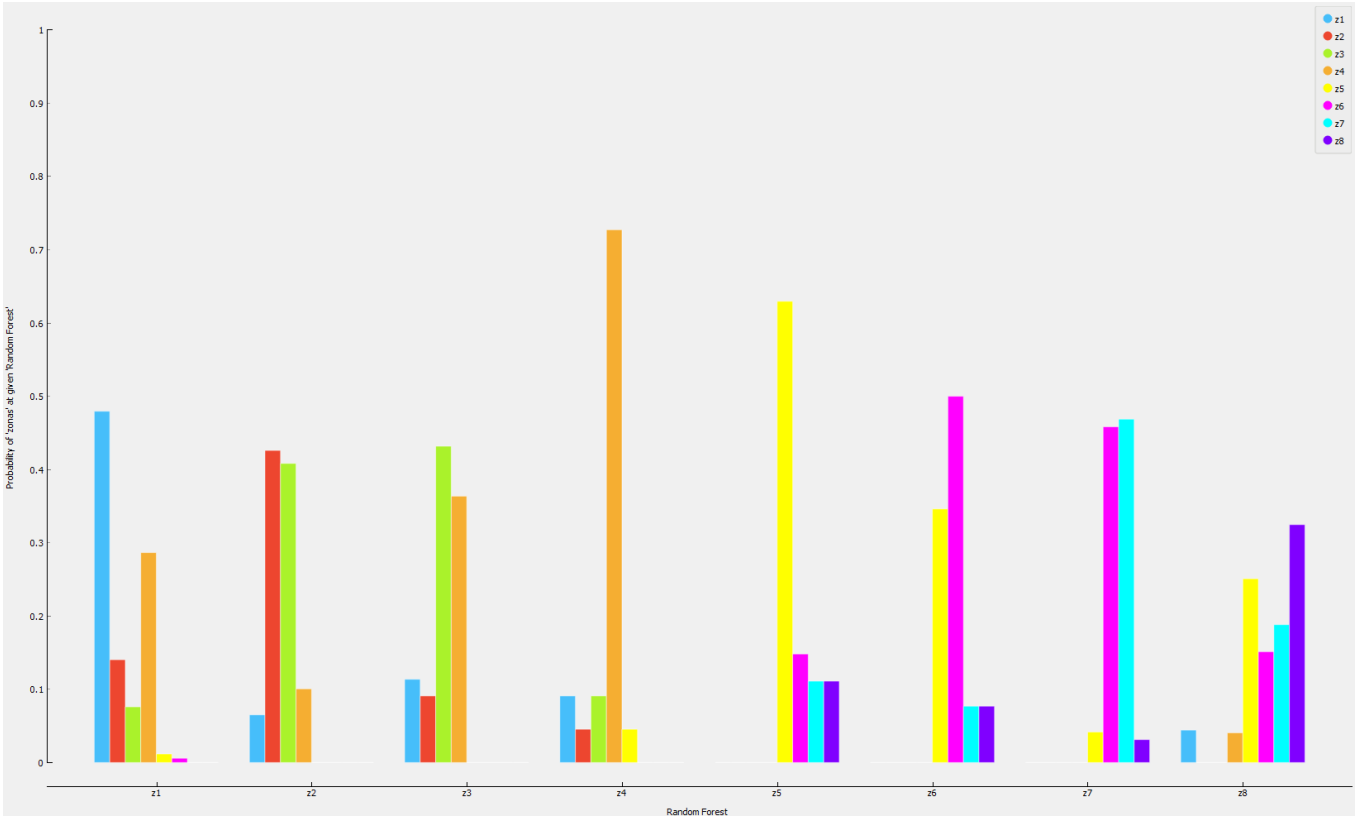


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	43.5 %	3.1 %	6.5 %	13.0 %	0.0 %	0.0 %	0.0 %	2.2 %	112
z2	15.2 %	45.4 %	15.2 %	4.3 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	11.7 %	46.2 %	34.8 %	4.3 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	26.9 %	5.4 %	43.5 %	69.6 %	0.0 %	0.0 %	0.0 %	2.7 %	109
z5	0.9 %	0.0 %	0.0 %	8.7 %	72.7 %	37.1 %	4.8 %	27.8 %	101
z6	0.4 %	0.0 %	0.0 %	0.0 %	9.1 %	45.7 %	46.0 %	12.1 %	103
z7	0.0 %	0.0 %	0.0 %	0.0 %	9.1 %	5.7 %	46.8 %	17.5 %	101
z8	1.3 %	0.0 %	0.0 %	0.0 %	9.1 %	11.4 %	2.4 %	37.7 %	96
∑	223	130	46	23	22	35	124	223	826

Fuente: Realización propia (2022).

Figura 82

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Random Forest

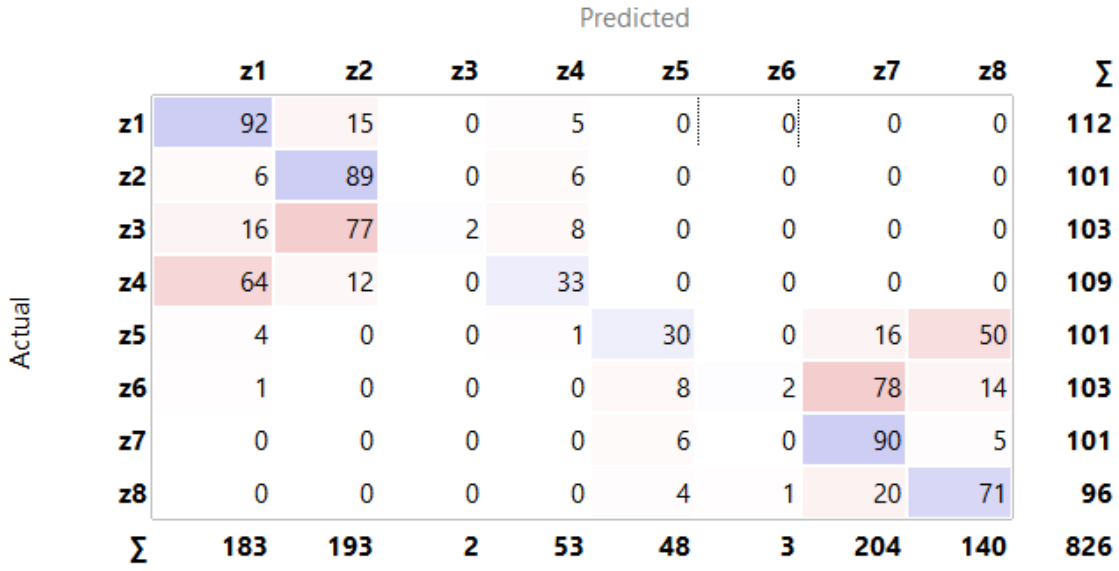


Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo KNN:

Figura 83

Matriz de confusión resultado de las pruebas con KNN

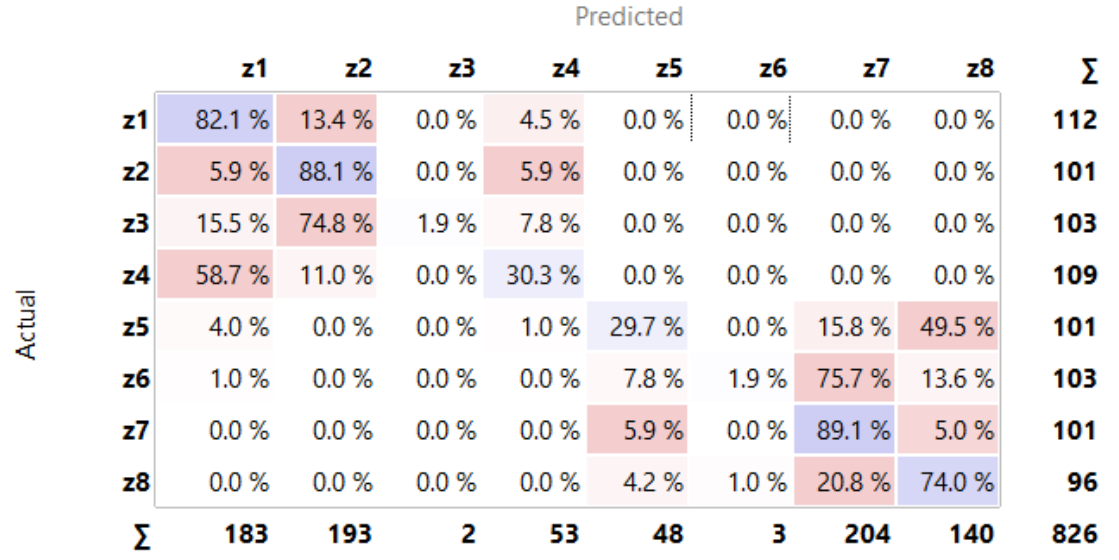


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	92	15	0	5	0	0	0	0	112
z2	6	89	0	6	0	0	0	0	101
z3	16	77	2	8	0	0	0	0	103
z4	64	12	0	33	0	0	0	0	109
z5	4	0	0	1	30	0	16	50	101
z6	1	0	0	0	8	2	78	14	103
z7	0	0	0	0	6	0	90	5	101
z8	0	0	0	0	4	1	20	71	96
∑	183	193	2	53	48	3	204	140	826

Fuente: Realización propia (2022).

Figura 84

Matriz de confusión en porcentaje resultado de las pruebas con KNN

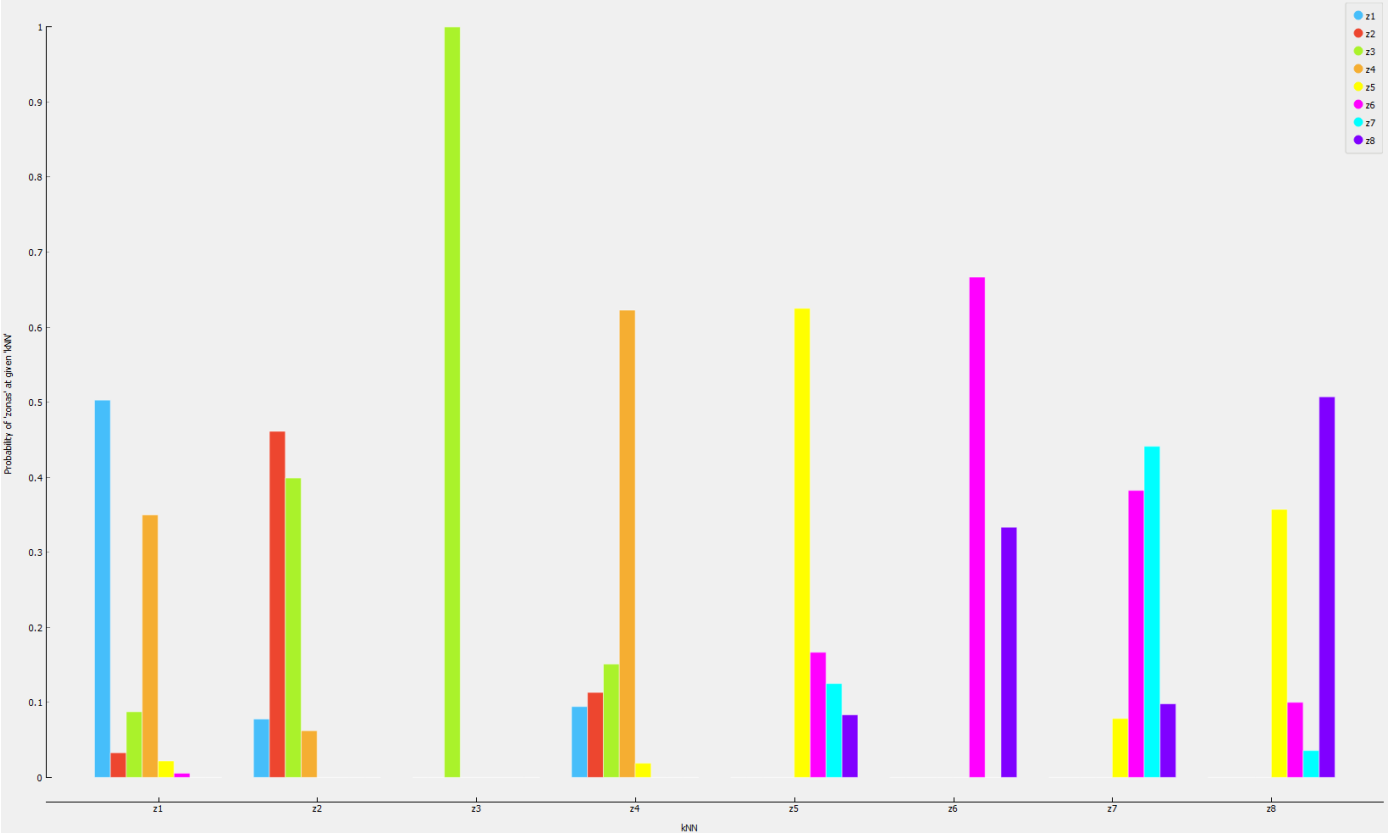


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	82.1 %	13.4 %	0.0 %	4.5 %	0.0 %	0.0 %	0.0 %	0.0 %	112
z2	5.9 %	88.1 %	0.0 %	5.9 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	15.5 %	74.8 %	1.9 %	7.8 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	58.7 %	11.0 %	0.0 %	30.3 %	0.0 %	0.0 %	0.0 %	0.0 %	109
z5	4.0 %	0.0 %	0.0 %	1.0 %	29.7 %	0.0 %	15.8 %	49.5 %	101
z6	1.0 %	0.0 %	0.0 %	0.0 %	7.8 %	1.9 %	75.7 %	13.6 %	103
z7	0.0 %	0.0 %	0.0 %	0.0 %	5.9 %	0.0 %	89.1 %	5.0 %	101
z8	0.0 %	0.0 %	0.0 %	0.0 %	4.2 %	1.0 %	20.8 %	74.0 %	96
∑	183	193	2	53	48	3	204	140	826

Fuente: Realización propia (2022).

Figura 85

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de KNN



Fuente: Realización propia (2022).

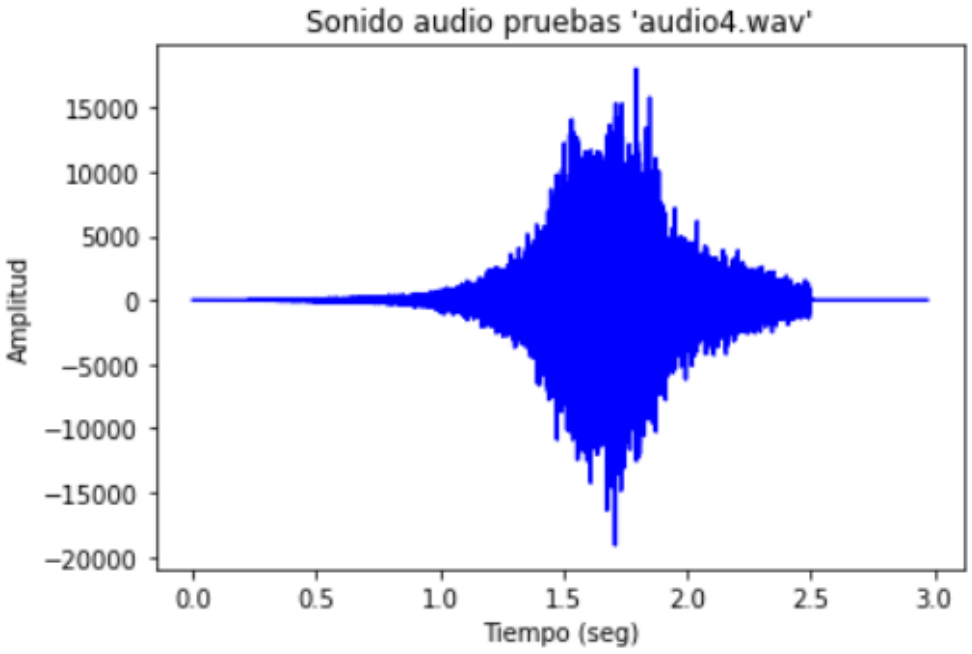
Prueba 3:

Nombre del audio: audio4.wav

Onda del sonido del audio de prueba:

Figura 86

Gráfico de la onda del sonido de prueba 'audio4.wav'



Fuente: Realización propia (2022).

Resultado de las pruebas:

Figura 87

Tabla de resultados de las pruebas

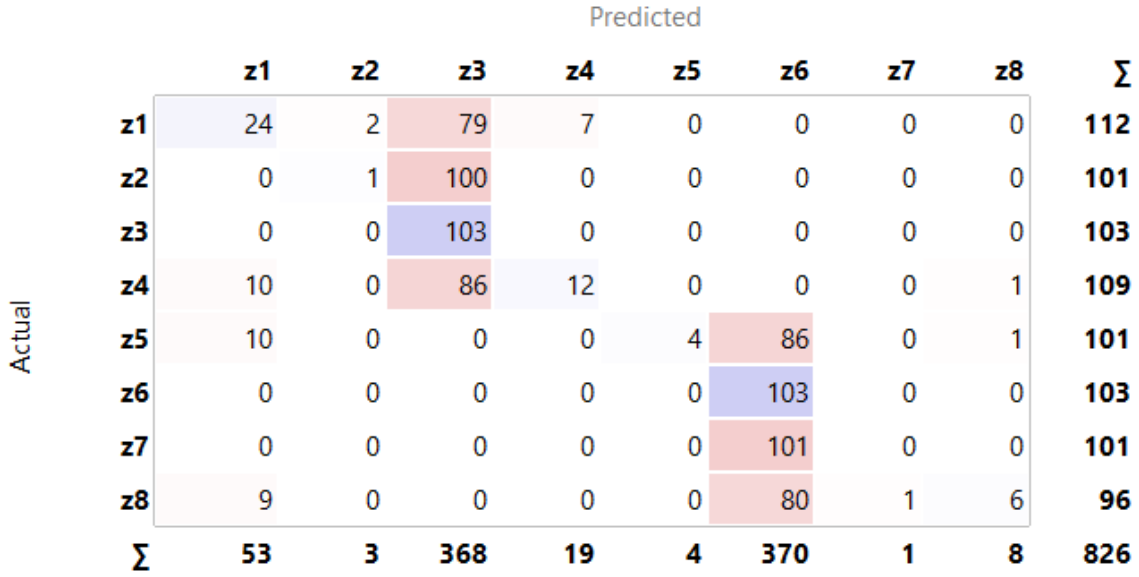
Modelo	Precisión	P tipo A	P tipo B
Neural Network	0,465	0,619	0,886
Random Forest	0,573	0,852	0,964
KNN	0,609	0,831	0,951

Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo Neural Network:

Figura 88

Matriz de confusión resultado de las pruebas con Red Neuronal

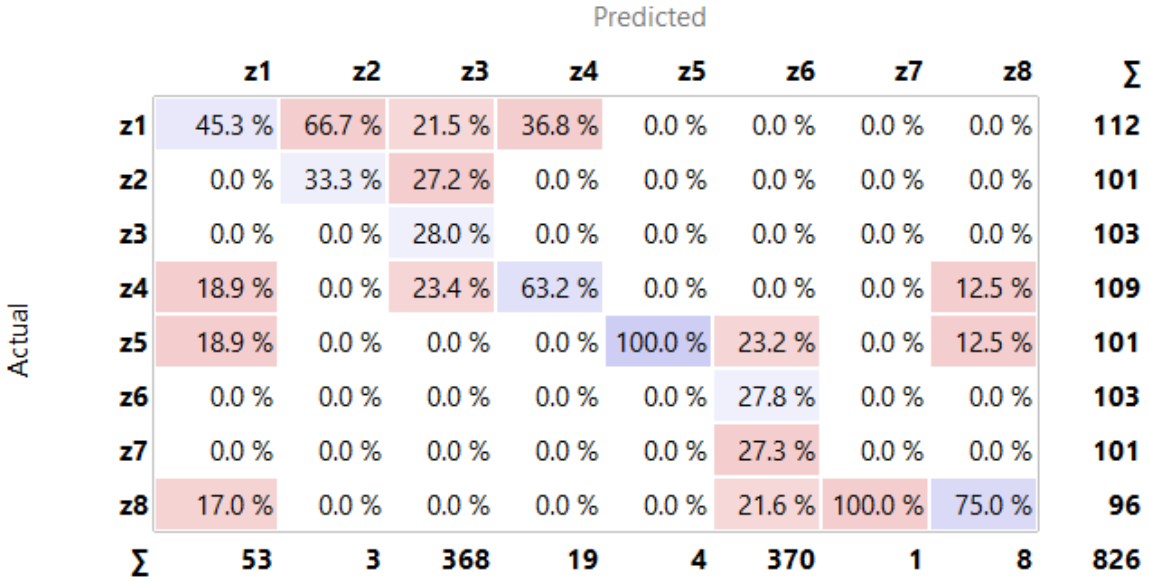


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	24	2	79	7	0	0	0	0	112
z2	0	1	100	0	0	0	0	0	101
z3	0	0	103	0	0	0	0	0	103
z4	10	0	86	12	0	0	0	1	109
z5	10	0	0	0	4	86	0	1	101
z6	0	0	0	0	0	103	0	0	103
z7	0	0	0	0	0	101	0	0	101
z8	9	0	0	0	0	80	1	6	96
∑	53	3	368	19	4	370	1	8	826

Fuente: Realización propia (2022).

Figura 89

Matriz de confusión en porcentaje resultado de las pruebas con Red Neuronal

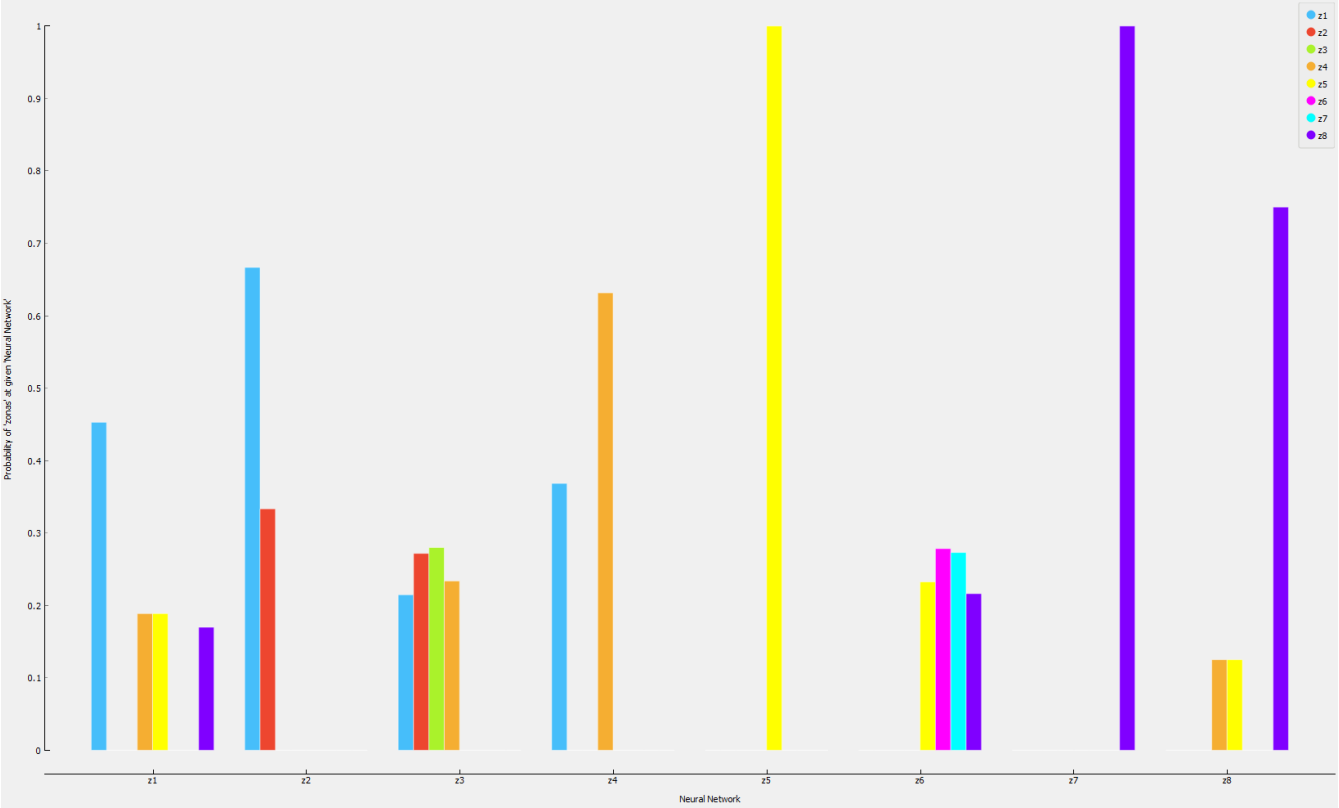


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	45.3 %	66.7 %	21.5 %	36.8 %	0.0 %	0.0 %	0.0 %	0.0 %	112
z2	0.0 %	33.3 %	27.2 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	0.0 %	0.0 %	28.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	18.9 %	0.0 %	23.4 %	63.2 %	0.0 %	0.0 %	0.0 %	12.5 %	109
z5	18.9 %	0.0 %	0.0 %	0.0 %	100.0 %	23.2 %	0.0 %	12.5 %	101
z6	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	27.8 %	0.0 %	0.0 %	103
z7	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	27.3 %	0.0 %	0.0 %	101
z8	17.0 %	0.0 %	0.0 %	0.0 %	0.0 %	21.6 %	100.0 %	75.0 %	96
∑	53	3	368	19	4	370	1	8	826

Fuente: Realización propia (2022).

Figura 90

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Red Neuronal

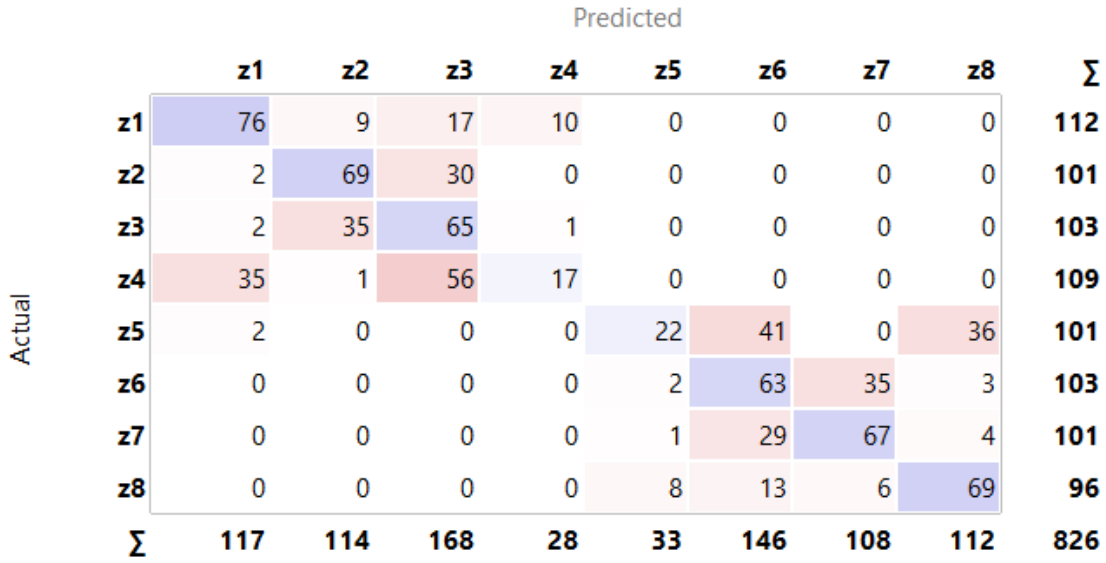


Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo Random Forest:

Figura 91

Matriz de confusión resultado de las pruebas con Random Forest

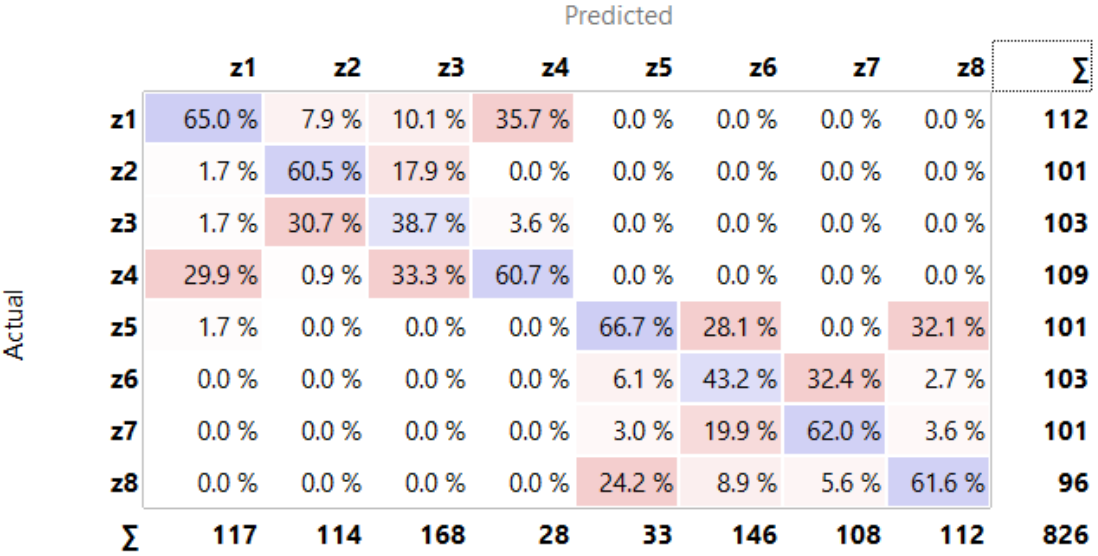


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	76	9	17	10	0	0	0	0	112
z2	2	69	30	0	0	0	0	0	101
z3	2	35	65	1	0	0	0	0	103
z4	35	1	56	17	0	0	0	0	109
z5	2	0	0	0	22	41	0	36	101
z6	0	0	0	0	2	63	35	3	103
z7	0	0	0	0	1	29	67	4	101
z8	0	0	0	0	8	13	6	69	96
∑	117	114	168	28	33	146	108	112	826

Fuente: Realización propia (2022).

Figura 92

Matriz de confusión resultado de las pruebas con Random Forest

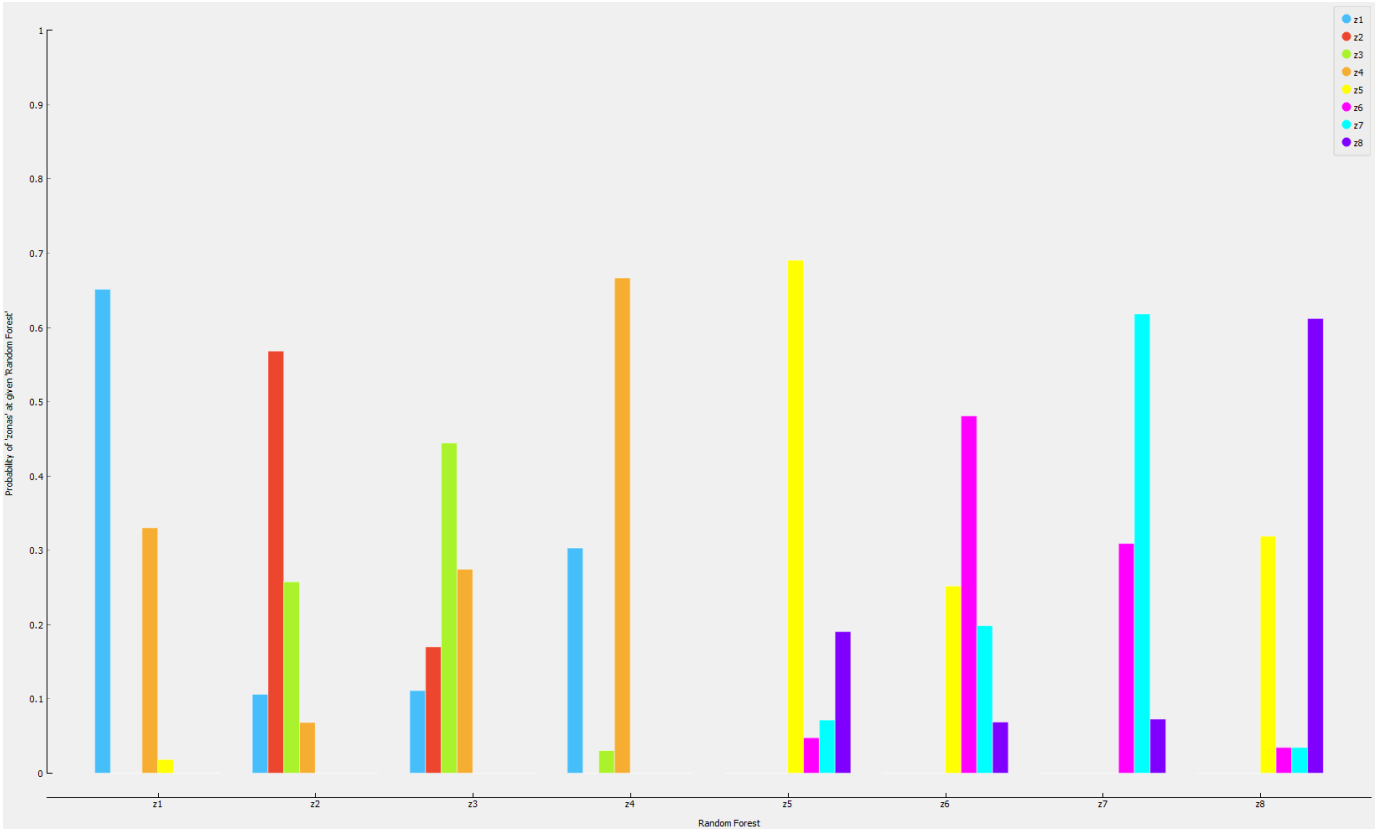


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	65.0 %	7.9 %	10.1 %	35.7 %	0.0 %	0.0 %	0.0 %	0.0 %	112
z2	1.7 %	60.5 %	17.9 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	1.7 %	30.7 %	38.7 %	3.6 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	29.9 %	0.9 %	33.3 %	60.7 %	0.0 %	0.0 %	0.0 %	0.0 %	109
z5	1.7 %	0.0 %	0.0 %	0.0 %	66.7 %	28.1 %	0.0 %	32.1 %	101
z6	0.0 %	0.0 %	0.0 %	0.0 %	6.1 %	43.2 %	32.4 %	2.7 %	103
z7	0.0 %	0.0 %	0.0 %	0.0 %	3.0 %	19.9 %	62.0 %	3.6 %	101
z8	0.0 %	0.0 %	0.0 %	0.0 %	24.2 %	8.9 %	5.6 %	61.6 %	96
∑	117	114	168	28	33	146	108	112	826

Fuente: Realización propia (2022).

Figura 93

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Random Forest

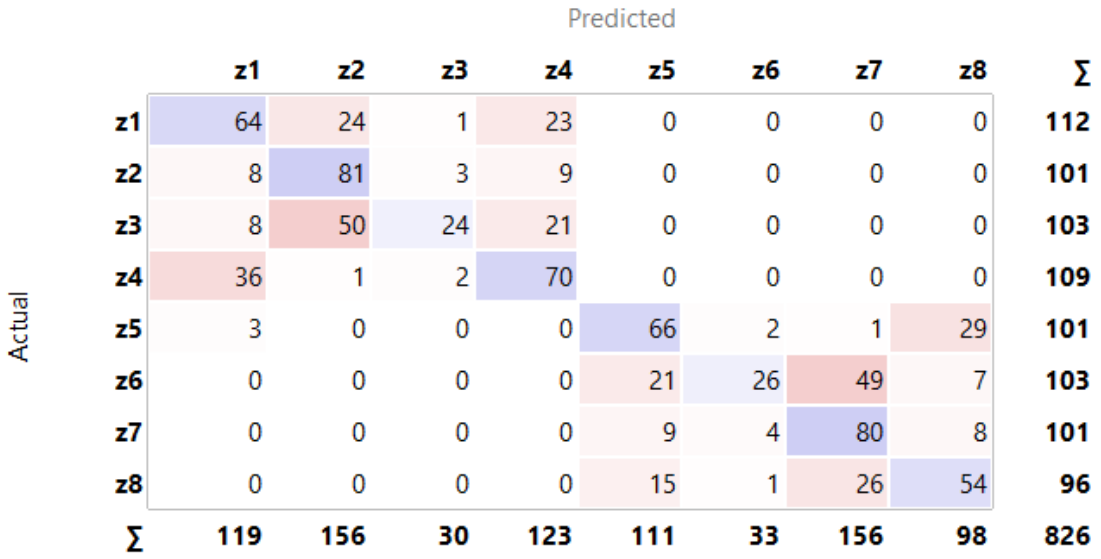


Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo KNN:

Figura 94

Matriz de confusión resultado de las pruebas con KNN

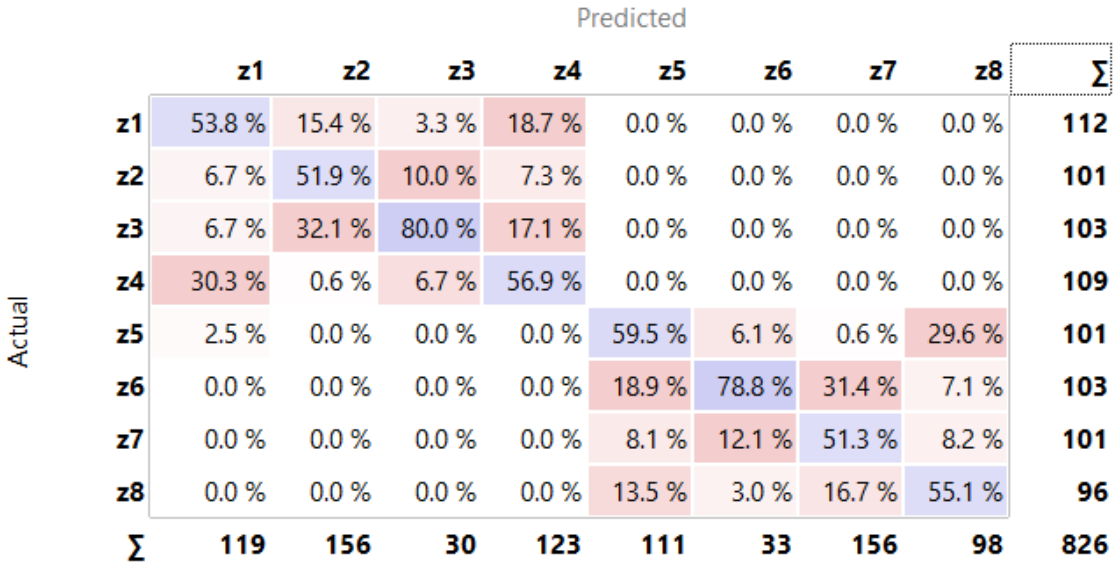


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	64	24	1	23	0	0	0	0	112
z2	8	81	3	9	0	0	0	0	101
z3	8	50	24	21	0	0	0	0	103
z4	36	1	2	70	0	0	0	0	109
z5	3	0	0	0	66	2	1	29	101
z6	0	0	0	0	21	26	49	7	103
z7	0	0	0	0	9	4	80	8	101
z8	0	0	0	0	15	1	26	54	96
∑	119	156	30	123	111	33	156	98	826

Fuente: Realización propia (2022).

Figura 95

Matriz de confusión resultado de las pruebas con KNN

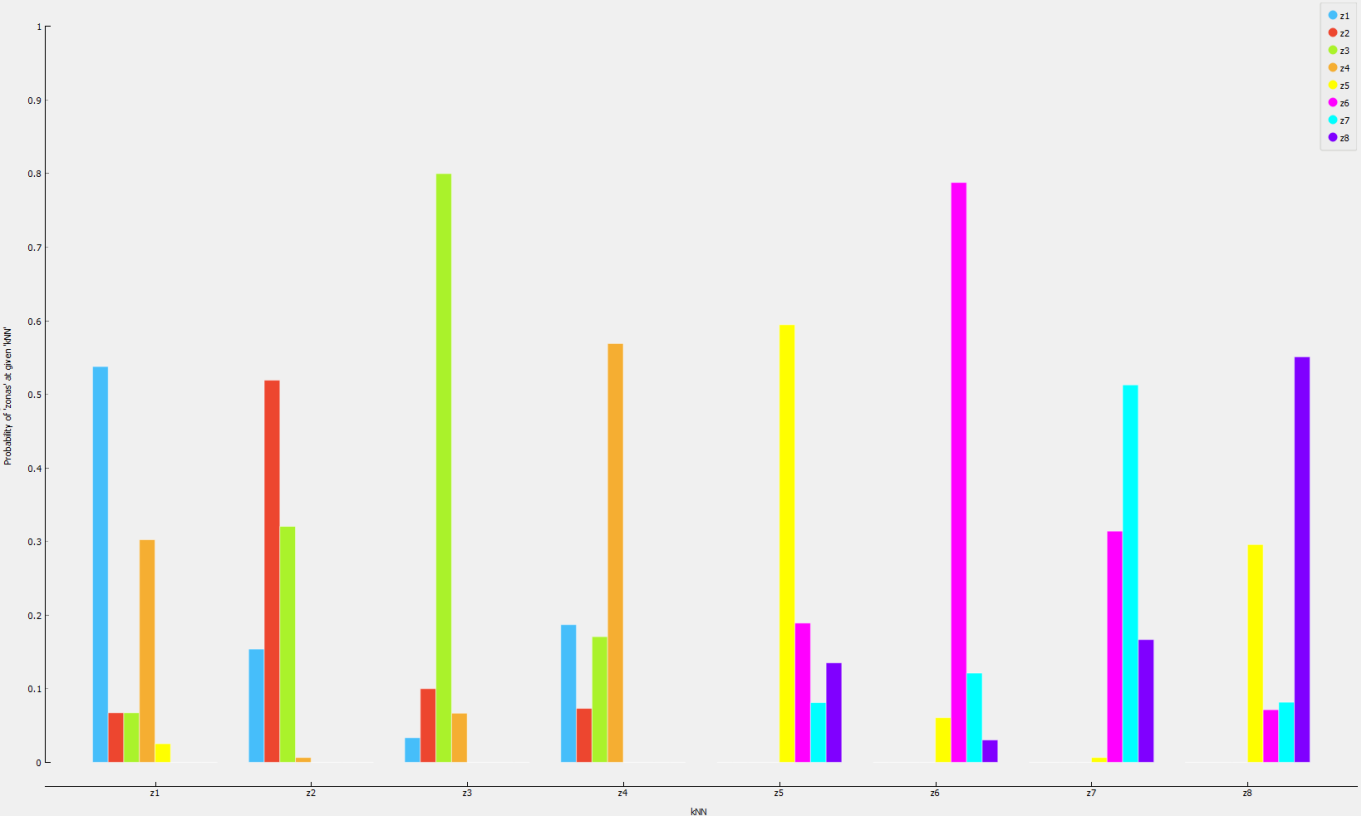


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	53.8 %	15.4 %	3.3 %	18.7 %	0.0 %	0.0 %	0.0 %	0.0 %	112
z2	6.7 %	51.9 %	10.0 %	7.3 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	6.7 %	32.1 %	80.0 %	17.1 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	30.3 %	0.6 %	6.7 %	56.9 %	0.0 %	0.0 %	0.0 %	0.0 %	109
z5	2.5 %	0.0 %	0.0 %	0.0 %	59.5 %	6.1 %	0.6 %	29.6 %	101
z6	0.0 %	0.0 %	0.0 %	0.0 %	18.9 %	78.8 %	31.4 %	7.1 %	103
z7	0.0 %	0.0 %	0.0 %	0.0 %	8.1 %	12.1 %	51.3 %	8.2 %	101
z8	0.0 %	0.0 %	0.0 %	0.0 %	13.5 %	3.0 %	16.7 %	55.1 %	96
∑	119	156	30	123	111	33	156	98	826

Fuente: Realización propia (2022).

Figura 96

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de KNN



Fuente: Realización propia (2022).

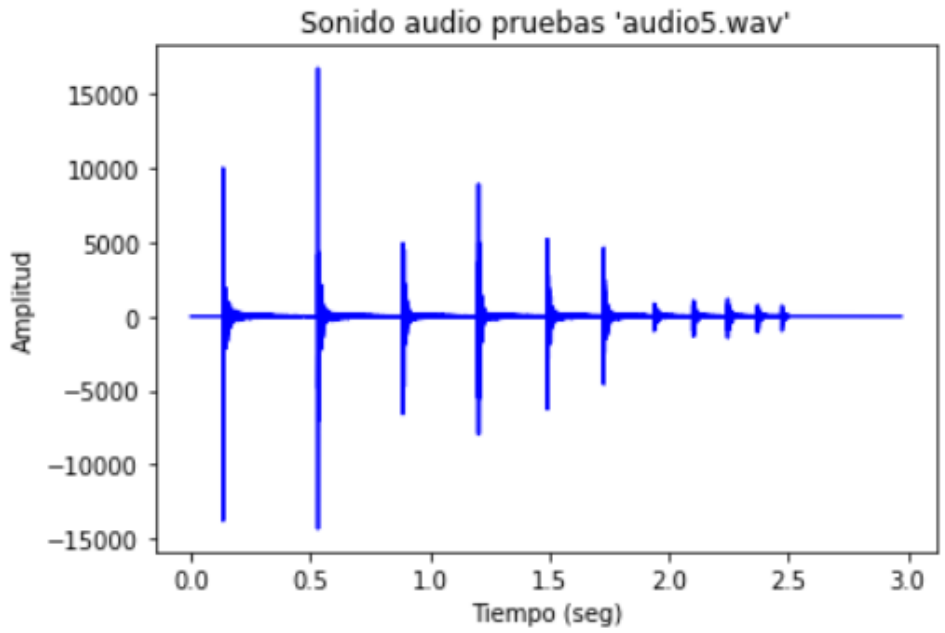
Prueba 4:

Nombre del audio: audio5.wav

Onda del sonido del audio de prueba:

Figura 97

Gráfico de la onda del sonido de prueba 'audio5.wav'



Fuente: Realización propia (2022).

Resultado de las pruebas:

Figura 98

Tabla de resultado de las pruebas

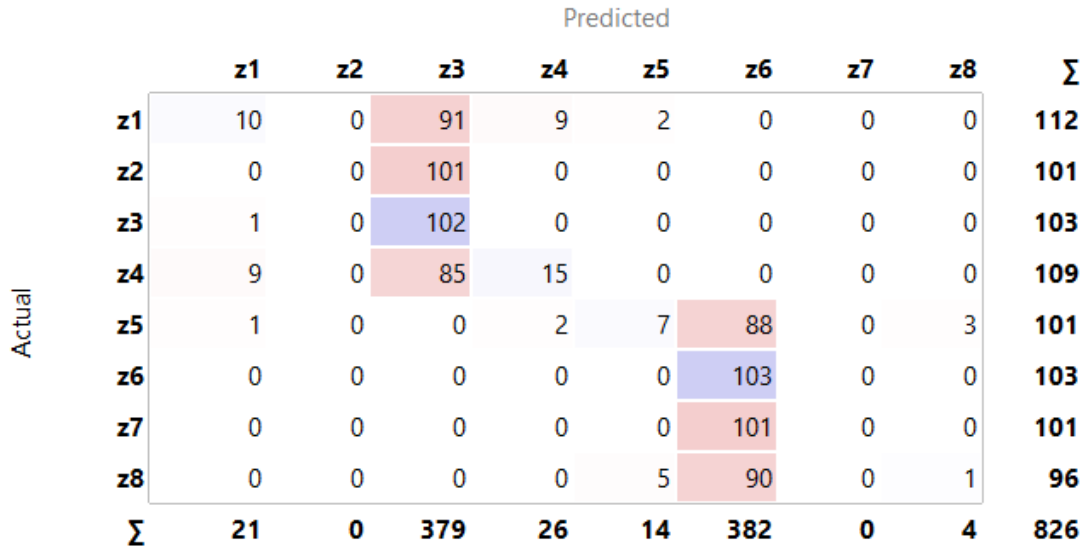
Modelo	Precisión	P tipo A	P tipo B
Neural Network	0,298	0,594	0,651
Random Forest	0,480	0,754	0,852
KNN	0,541	0,814	0,939

Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo Neural Network:

Figura 99

Matriz de confusión resultado de las pruebas con Red Neuronal

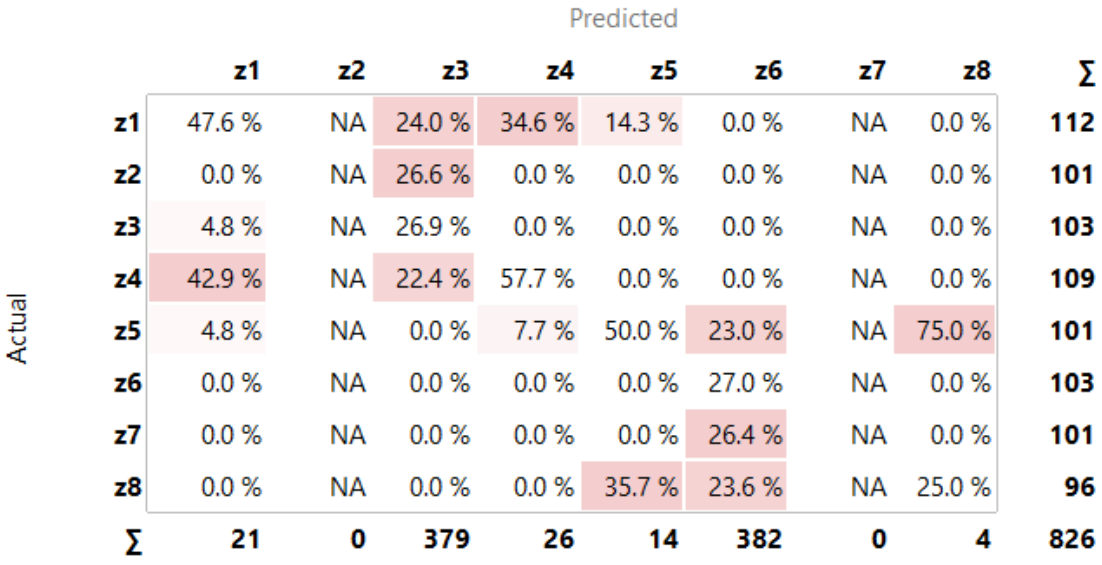


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	10	0	91	9	2	0	0	0	112
z2	0	0	101	0	0	0	0	0	101
z3	1	0	102	0	0	0	0	0	103
z4	9	0	85	15	0	0	0	0	109
z5	1	0	0	2	7	88	0	3	101
z6	0	0	0	0	0	103	0	0	103
z7	0	0	0	0	0	101	0	0	101
z8	0	0	0	0	5	90	0	1	96
∑	21	0	379	26	14	382	0	4	826

Fuente: Realización propia (2022).

Figura 100

Matriz de confusión en porcentaje resultado de las pruebas con Red Neuronal

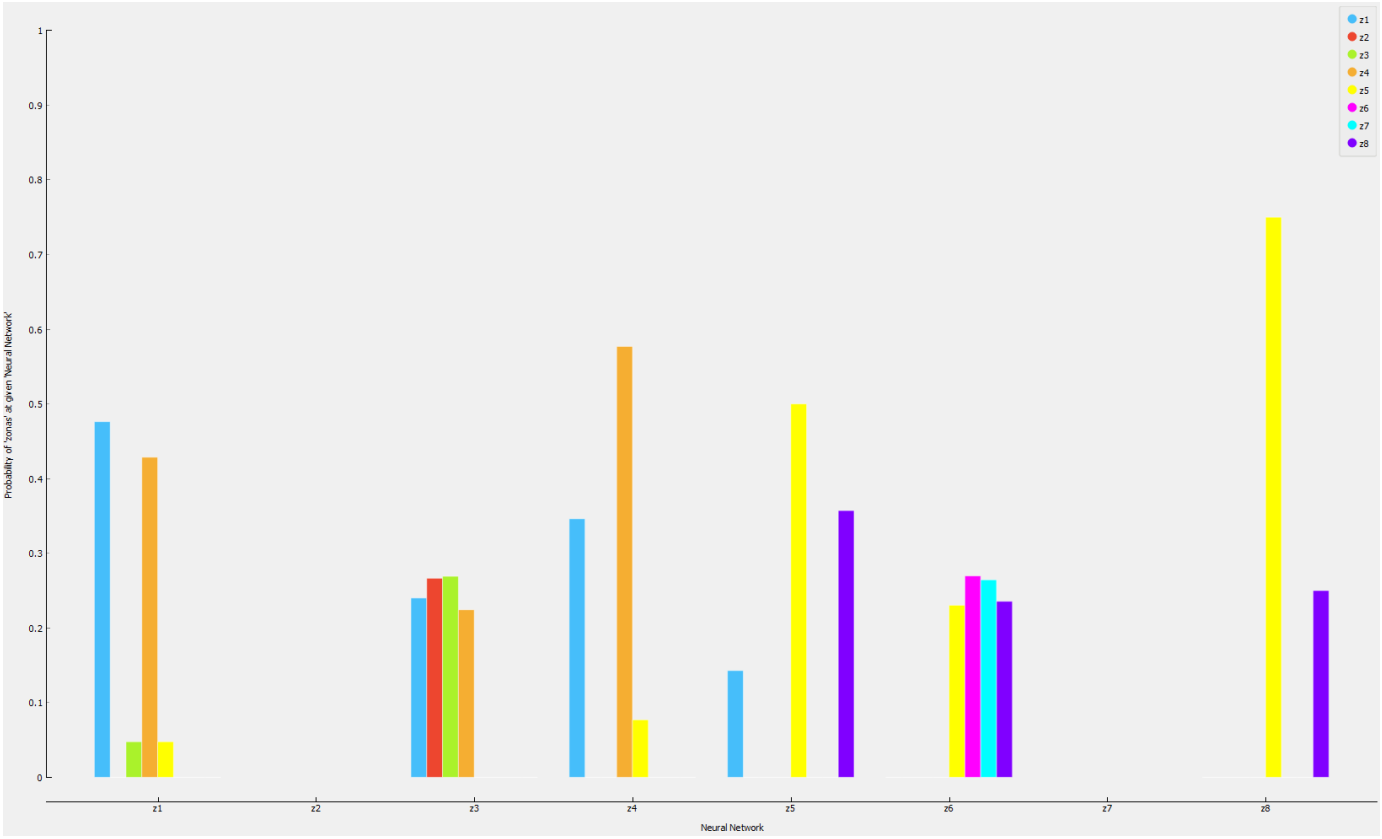


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	47.6 %	NA	24.0 %	34.6 %	14.3 %	0.0 %	NA	0.0 %	112
z2	0.0 %	NA	26.6 %	0.0 %	0.0 %	0.0 %	NA	0.0 %	101
z3	4.8 %	NA	26.9 %	0.0 %	0.0 %	0.0 %	NA	0.0 %	103
z4	42.9 %	NA	22.4 %	57.7 %	0.0 %	0.0 %	NA	0.0 %	109
z5	4.8 %	NA	0.0 %	7.7 %	50.0 %	23.0 %	NA	75.0 %	101
z6	0.0 %	NA	0.0 %	0.0 %	0.0 %	27.0 %	NA	0.0 %	103
z7	0.0 %	NA	0.0 %	0.0 %	0.0 %	26.4 %	NA	0.0 %	101
z8	0.0 %	NA	0.0 %	0.0 %	35.7 %	23.6 %	NA	25.0 %	96
∑	21	0	379	26	14	382	0	4	826

Fuente: Realización propia (2022).

Figura 101

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Red Neuronal

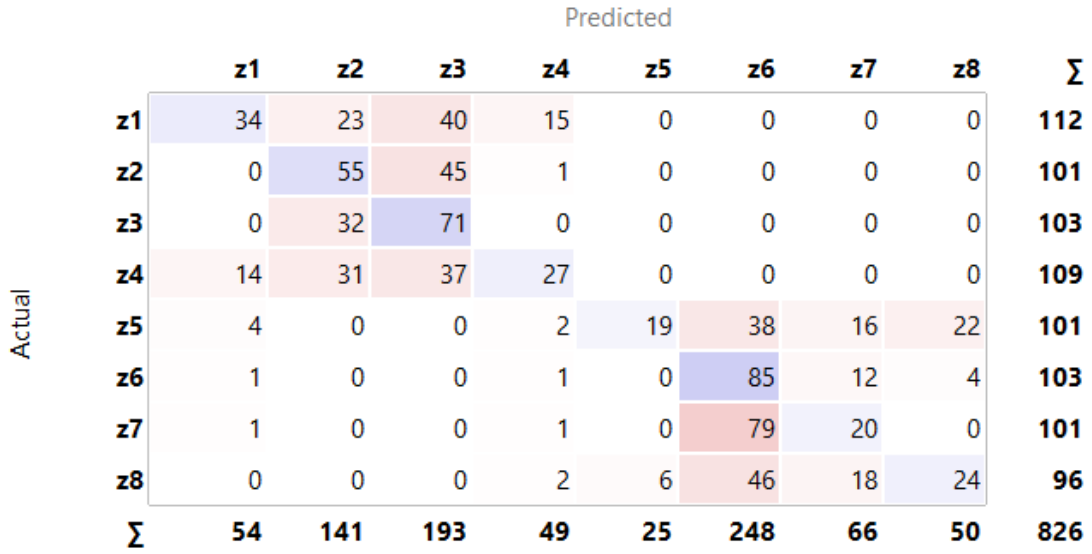


Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo Random Forest:

Figura 102

Matriz de confusión resultado de las pruebas con Random Forest

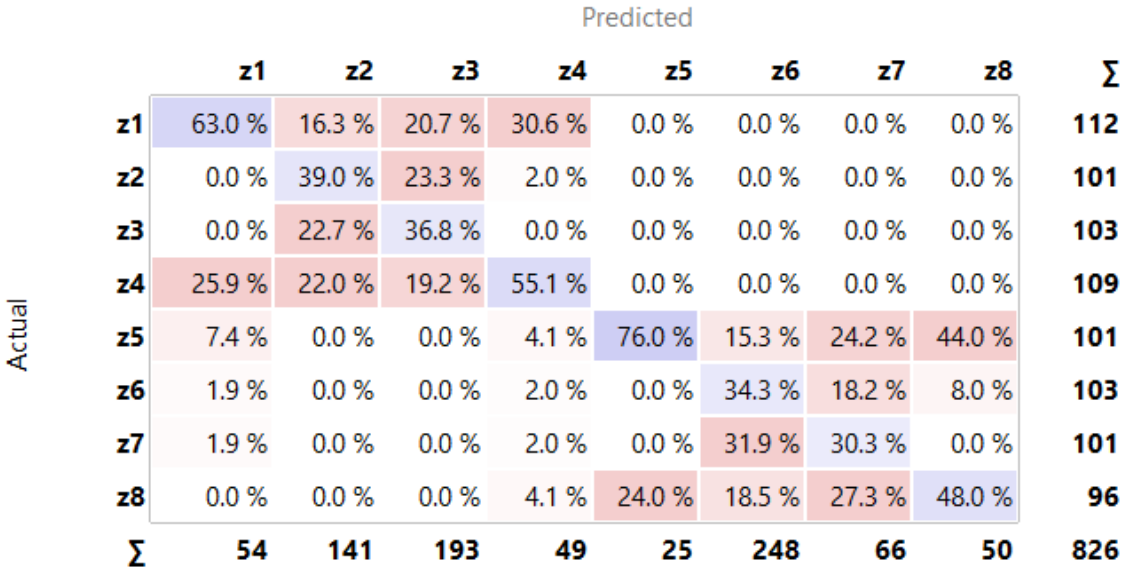


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	34	23	40	15	0	0	0	0	112
z2	0	55	45	1	0	0	0	0	101
z3	0	32	71	0	0	0	0	0	103
z4	14	31	37	27	0	0	0	0	109
z5	4	0	0	2	19	38	16	22	101
z6	1	0	0	1	0	85	12	4	103
z7	1	0	0	1	0	79	20	0	101
z8	0	0	0	2	6	46	18	24	96
∑	54	141	193	49	25	248	66	50	826

Fuente: Realización propia (2022).

Figura 103

Matriz de confusión en porcentaje resultado de las pruebas con Random Forest

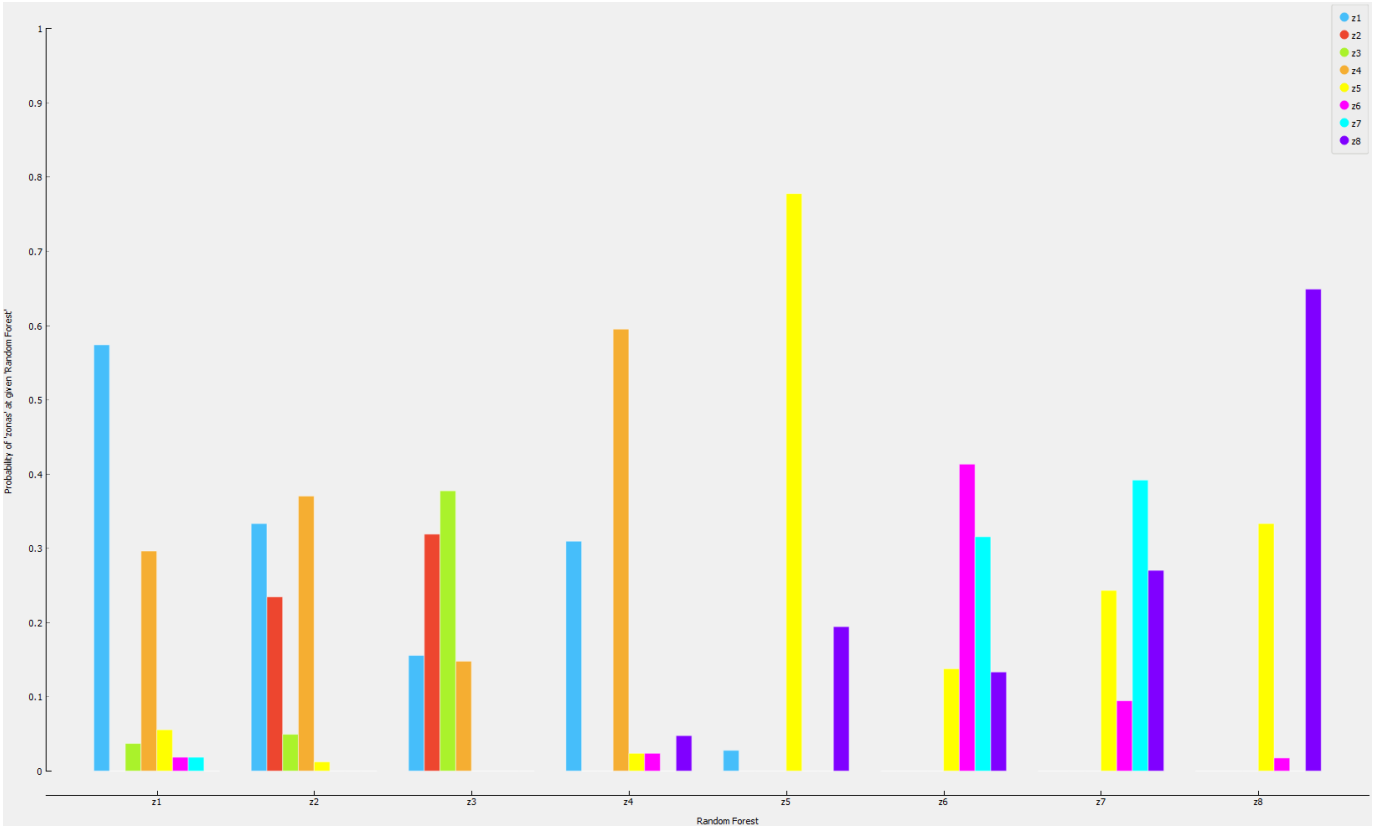


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	63.0 %	16.3 %	20.7 %	30.6 %	0.0 %	0.0 %	0.0 %	0.0 %	112
z2	0.0 %	39.0 %	23.3 %	2.0 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	0.0 %	22.7 %	36.8 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	25.9 %	22.0 %	19.2 %	55.1 %	0.0 %	0.0 %	0.0 %	0.0 %	109
z5	7.4 %	0.0 %	0.0 %	4.1 %	76.0 %	15.3 %	24.2 %	44.0 %	101
z6	1.9 %	0.0 %	0.0 %	2.0 %	0.0 %	34.3 %	18.2 %	8.0 %	103
z7	1.9 %	0.0 %	0.0 %	2.0 %	0.0 %	31.9 %	30.3 %	0.0 %	101
z8	0.0 %	0.0 %	0.0 %	4.1 %	24.0 %	18.5 %	27.3 %	48.0 %	96
∑	54	141	193	49	25	248	66	50	826

Fuente: Realización propia (2022).

Figura 104

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de Random Forest

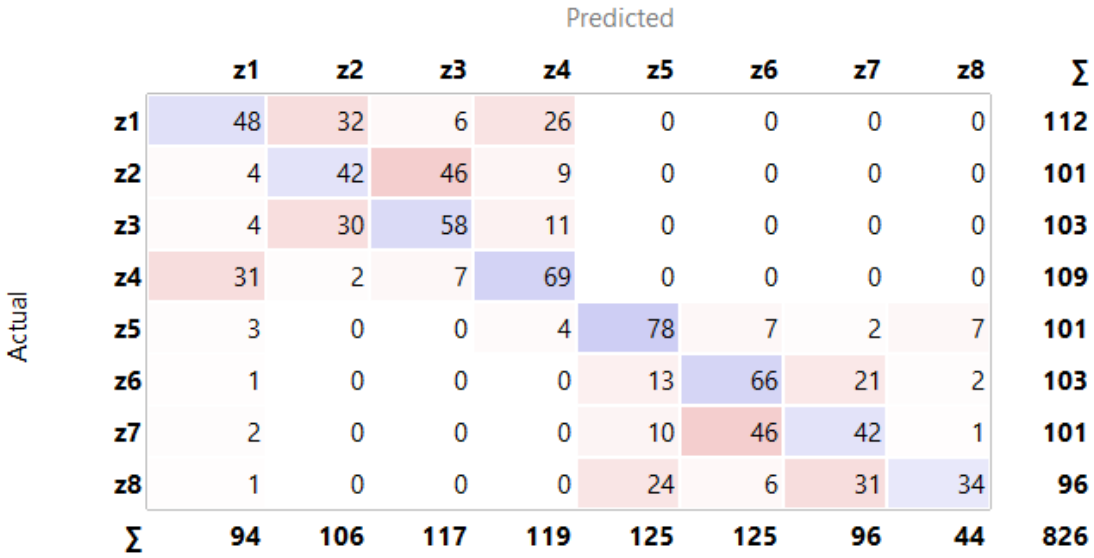


Fuente: Realización propia (2022).

Matriz de confusión generada por el modelo KNN:

Figura 105

Matriz de confusión resultado de las pruebas con KNN

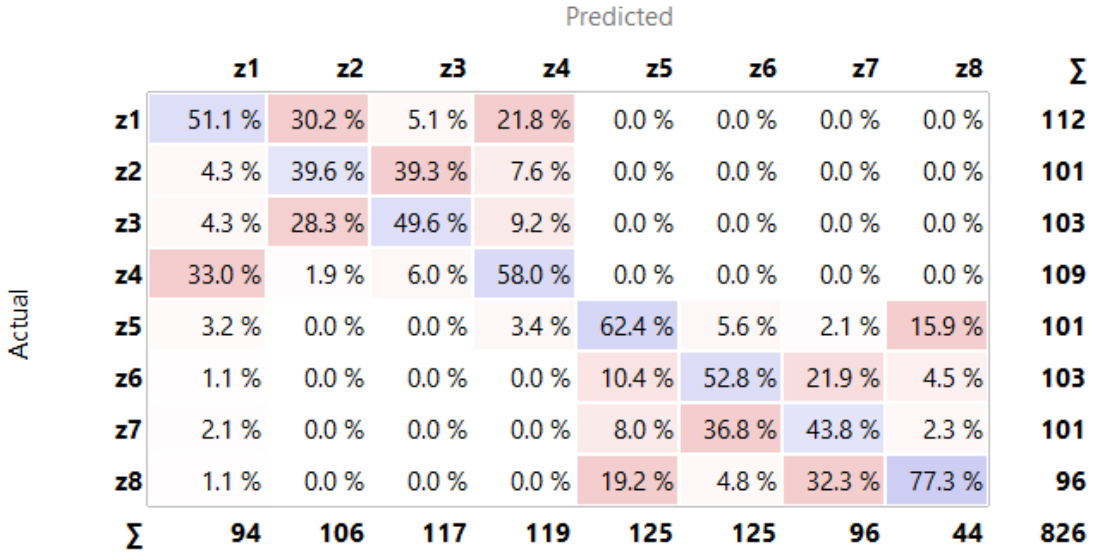


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	48	32	6	26	0	0	0	0	112
z2	4	42	46	9	0	0	0	0	101
z3	4	30	58	11	0	0	0	0	103
z4	31	2	7	69	0	0	0	0	109
z5	3	0	0	4	78	7	2	7	101
z6	1	0	0	0	13	66	21	2	103
z7	2	0	0	0	10	46	42	1	101
z8	1	0	0	0	24	6	31	34	96
∑	94	106	117	119	125	125	96	44	826

Fuente: Realización propia (2022).

Figura 106

Matriz de confusión en porcentaje resultado de las pruebas con KNN

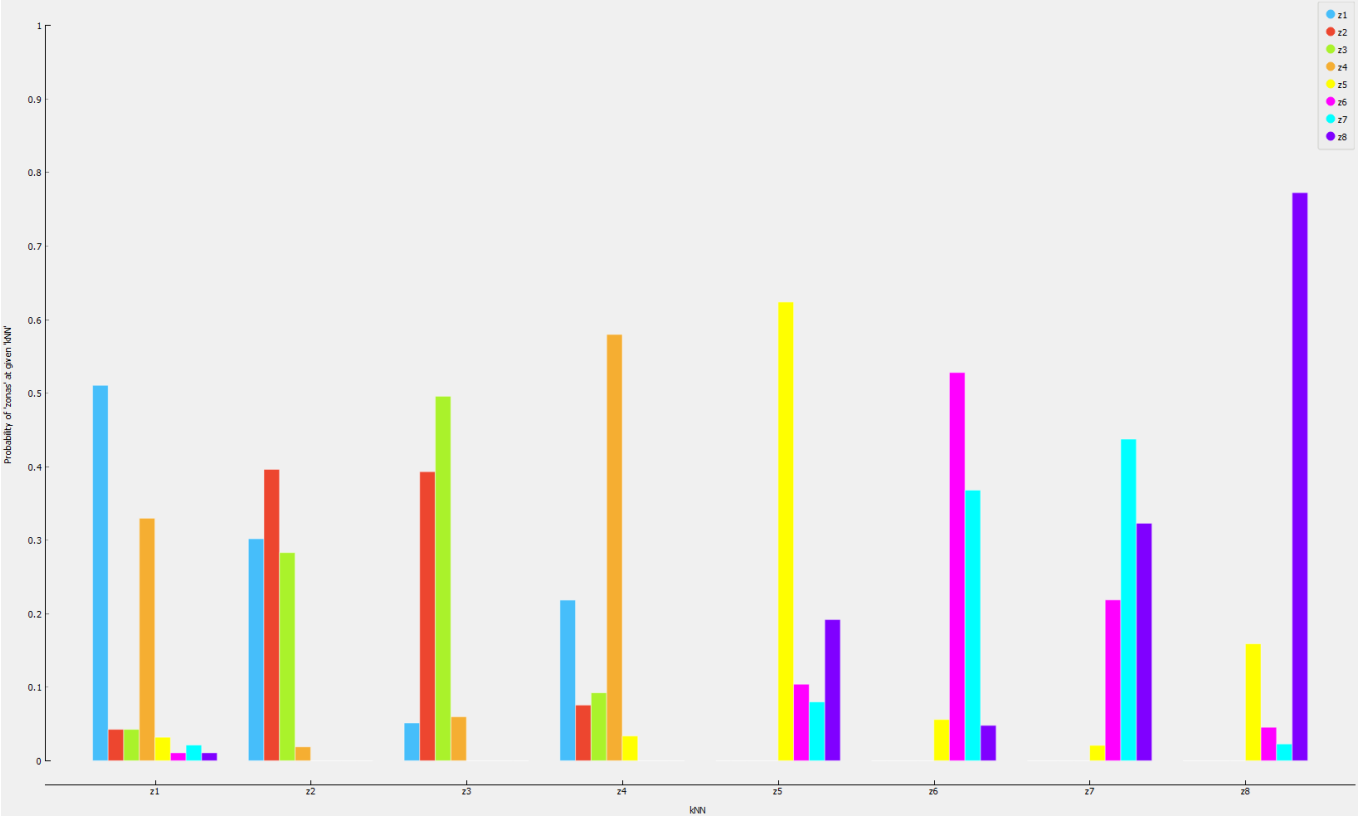


Actual \ Predicted	z1	z2	z3	z4	z5	z6	z7	z8	∑
z1	51.1 %	30.2 %	5.1 %	21.8 %	0.0 %	0.0 %	0.0 %	0.0 %	112
z2	4.3 %	39.6 %	39.3 %	7.6 %	0.0 %	0.0 %	0.0 %	0.0 %	101
z3	4.3 %	28.3 %	49.6 %	9.2 %	0.0 %	0.0 %	0.0 %	0.0 %	103
z4	33.0 %	1.9 %	6.0 %	58.0 %	0.0 %	0.0 %	0.0 %	0.0 %	109
z5	3.2 %	0.0 %	0.0 %	3.4 %	62.4 %	5.6 %	2.1 %	15.9 %	101
z6	1.1 %	0.0 %	0.0 %	0.0 %	10.4 %	52.8 %	21.9 %	4.5 %	103
z7	2.1 %	0.0 %	0.0 %	0.0 %	8.0 %	36.8 %	43.8 %	2.3 %	101
z8	1.1 %	0.0 %	0.0 %	0.0 %	19.2 %	4.8 %	32.3 %	77.3 %	96
∑	94	106	117	119	125	125	96	44	826

Fuente: Realización propia (2022).

Figura 107

Gráfico de distribución en porcentaje de las zonas seleccionadas por el modelo de KNN



Fuente: Realización propia (2022).

De esta manera se pueden resumir los resultados en la tabla de la siguiente figura:

Figura 108

Tabla de resultado de las pruebas con los modelos

Modelo	Prueba	Precisión	P tipo A	P tipo B
Neural Network	Prueba 1	0,532	0,772	0,975
	Prueba 2	0,382	0,514	0,82
	Prueba 3	0,465	0,619	0,886
	Prueba 4	0,298	0,594	0,651
	Promedio	0,419	0,625	0,833
Random Forest	Prueba 1	0,502	0,846	0,926
	Prueba 2	0,497	0,733	0,898
	Prueba 3	0,573	0,852	0,964
	Prueba 4	0,480	0,754	0,852
	Promedio	0,513	0,796	0,910
KNN	Prueba 1	0,632	0,873	0,944
	Prueba 2	0,604	0,812	0,882
	Prueba 3	0,609	0,831	0,951
	Prueba 4	0,541	0,814	0,939
	Promedio	0,597	0,833	0,929

Fuente: Realización propia (2022).

5.2. DISCUSIÓN DE RESULTADOS

Una vez realizado el entrenamiento y las respectivas pruebas con los tres modelos iniciales se encontró que el modelo de KN- Neighbors obtuvo un valor de precisión insuficiente para los requerimientos del proyecto. Sin embargo, en las pruebas con audios desconocidos para el modelo su precisión con respecto a los otros modelos fue superior. Esto es muestra de que le modelo KN-Neighbors tuvo una mayor capacidad de adaptarse a otros audios.

En el caso del modelo Neural Network en el entrenamiento dio una precisión del 77,7%. Sin embargo, a la hora de realizar las pruebas de validación los valores obtenidos promedio de todas las pruebas fueron del 41,9%, con errores tipo A 62,5% y errores tipo B 83,3%. Mientras que para el modelo de Random Forest en el entrenamiento se obtuvo un valor del 74,4% de precisión. Ahora bien, en las pruebas de validación este modelo obtuvo un promedio de precisión del 51,3%, con errores tipo A 79,6% y errores tipo B 91,1%. Finalmente, el modelo de K-N Neighbors en el entrenamiento obtuvo una precisión 65,5% mientras que en las pruebas de validación obtuvo 59,7%, con errores tipo A 83,3 % y errores tipo B 92,9%.

Hay que comentar también que en el caso del uso de recursos computacionales el modelo de Neural Network es más exigente e incluso el tiempo que le toma en el entrenamiento y realizar las validaciones comparado con el modelo de Random Forest y K-N Neighbors es 15x mayor.

Por lo resultados de las pruebas de validación con audios completamente desconocidos para los modelos podemos decir que tanto los modelos Neural Network y Random Forest sufrieron un sobre entrenamiento, esto debido a que en el entrenamiento tuvieron una precisión alta mientras que en las pruebas de validación los valores cayeron significativamente. Mientras

que el modelo de KN-Neighbors obtuvo una precisión más cercana a la que obtuvo en el entrenamiento. Este modelo es el más uniforme en cuanto a la precisión ya que obtuvo valores similares con todas las pruebas de validación.

Como se observa ninguno de los modelos cumple al menos un 70% de precisión después de las pruebas de validación sin tomar en cuenta los errores tipo A y B. Sin embargo, en este punto se debe apreciar que los modelos tienen una gran capacidad de reconocer la zona de procedencia del sonido ya que a diferencia de clasificar los audios de forma aleatoria 1/8 = 12,5% de precisión, los resultados obtenidos son bastantes superiores en los tres modelos. A demás hay que tener en cuenta que el problema del cono de confusión tuvo una influencia importante en los resultados, esto debido a que para las personas este problema se soluciona moviendo la cabeza un poco, acción que los modelos no la pueden hacer. En este sentido si tomamos en cuenta las veces que los modelos pueden caer en este problema el valor de la precisión aumenta en cada modelo. En este punto el modelo Neural Network obtuvo un valor de 62,5% de precisión que aún sigue siendo bajo para lo requerido, por tal motivo este modelo queda descartado.

Los modelos Random Forest y K-N Neighbors terminan con una precisión de 79,6% y 83,3% respectivamente, valores superiores a lo requerido. A demás si tomamos en cuenta los errores tipo B en que los modelos se confunden entre zonas por su cercanía, los valores de precisión aumentan a 91% y 92,9% respectivamente. Definiendo que para este proyecto solo se aceptará errores de tipo A y con los antecedentes podemos seleccionar el modelo de KN-Neighbors como el modelo óptimo y funcional para este proyecto con una precisión de 83,3%. Esto debido a que, aunque los valores son muy cercanos entre ambos modelos el de Random Forest es ligeramente inferior a los de KN-Neighbors.

6. CONCLUSIONES Y RECOMENDACIONES

Como conclusión se puede afirmar que se obtuvo un modelo de manera exitosa, el mismo que cumple con los requerimientos para solventar el problema. El tipo de modelo seleccionado fue el de KN-Neighbors, modelo que alcanzó con una precisión del 83,3%.

Para seleccionar el modelo más adecuado se tuvo que analizar entre varios modelos de manera teórica como práctica. Los modelos que se utilizó para este ejercicio fueron KN-Neighbors, Random Forest y Neural Network. Para la parte práctica se realizó el entrenamiento de cada uno de estos modelos. En este entrenamiento se fue realizando pruebas, cambiando el valor de los parámetros de cada modelo con el fin de optimizarlos lo mejor posible. De estos modelos el de KN-Neighbors fue el menos preciso con un valor de 65,5% de precisión. A partir de aquí se realizó las pruebas de validación con los tres modelos en las que este mismo modelo KN-Neighbors tuvo mejor respuesta con un valor promedio de precisión del 83,3% vs el de 79,6% del modelo Random Forest y el de 62,5% del modelo Neural Network.

Para el tema del dataset de entrenamiento se generó varios audios estéreo a partir de un audio mono auricular. Este proceso se lo realizó basándonos en la creación de una esfera formada por los ángulos de azimut y de elevación. De esta forma se generó puntos uniformemente distribuidos con lo que cada uno tiene su dirección de procedencia y por ende diferentes HRTF’s, Cada uno de estos audios no se repite con ningún otro. A partir de aquí se tuvo que extraer las características de cada audio y se formó el dataset en un archivo tipo CVS. También se generó audios para realizar las pruebas de validación. Para el dataset de entrenamiento se creó 826 audios estéreo y para las pruebas de validación se realizó de igual manera 826 audios estéreo por cada audio de prueba, al haber realizado cuatro pruebas en total se

generó 3 304 audios. Estos audios sumados a los audios de entrenamiento dan un total de 4 130 audios estéreos generados para este proyecto.

Se encontró gran influencia del problema del cono de confusión en los resultados obtenidos en todos los modelos. Por lo que se generó dos tipos de errores siendo los del tipo A aquellos que el modelo selecciona erróneamente la zona de procedencia de un audio debido al problema del cono de confusión y los del tipo B que corresponden aquellos que el modelo selecciona erróneamente la zona de procedencia de un audio debido a que se encuentra adyacente a la zona correcta. Sin embargo, para la precisión del modelo de este proyecto solo se acepta hasta los errores del tipo A.

Como recomendación para este tipo de proyectos se debe hacer hincapié en el tema de realizar una limpieza y control de la data antes de realizar el entrenamiento. Esto debido a que la mayor parte del proyecto debe enfocarse a esta parte, pues según la data que ingrese al modelo se tendrá la data de salida. Es decir, si se ingresa datos erróneos o mal estructurados, es posible que el modelo arroje valores incorrectos. Este hubiera sido el caso si no se hubieran revisado los valores atípicos en este proyecto, de haber mantenido estos valores es muy probable que no se hubiera obtenido el valor de precisión como el que se obtuvo.

7. BIBLIOGRAFÍA

Random Forest. (2022). *Analytics Lane*. Obtenido de https://www.analyticslane.com/2019/05/20/random-forest/

Artola Moreno, A. (2019). *Clasificación de imágenes usando redes neuronales convolucionales en Python.*

Berlanga, V.; Rubio Hurtado, M. J., & Vilà Baños, R. (2013). Cómo aplicar árboles de decisión en SPSS. REIRE. , vol. 6, num. 1,. *Revista d'Innovació i Recerca en Educació*, 65-79.

Cardelus, D., & Lorenzo, L. (2019). *Localización de fugas en terreno con machine learning.* Asociación Española de Abastecimientos de Agua y Saneamiento.

Cetta, P. (2003). *Localización espacial del sonido.* Obtenido de http://www. biopus. com. ar/matias/materias/apuntes/cetta_espacializacion_psicoac ustica. pdf

Cetta, P. C. (2007). *Un modelo para la simulación del espacio en música. Educa.* Buenos Aires: Educa.

Costa, J; Madrid, J A; Zamora, S. (1993). *Manual de clases prácticas de fisiología animal* (Vol. 28). Madrid: EDITUM.

Fraga Domingo, J. (2019). *CLASIFICACIÓN ACÚSTICA DE SALAS MEDIANTE.* Valencia: Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València.

GANIC, A. (2020). *DISEÑO Y DESARROLLO DE UN EQUIPO ELECTRÓNICO PARA LA MEDIDA DEL NIVEL DE PRESIÓN SONORA (Grado en Ingeniería Electrónica Industrial y Automática).* UNIVERSITAT POLITÈCNICA DE VALÈNCIA.

Gardner, B., & Martin, K. (1994). *Medidas HRTF de un micrófono de cabeza simulada KEMAR.*

Haykin, S. (2008). *Neural networks and learning machines (3rd ed.).* Ontorio Canadá: Pearson.

Jaramillo, A. M. (2007). *Acústica: la ciencia del sonido.* ITM.

Kramer, O. (2013). *K-nearest neighbors. In Dimensionality reduction with unsupervised nearest neighbors (pp. 13-23.* Berlin: Springer Heidelberg.

Lazcano Enciso , A. (2006). *Estudio de percepción sonora y valoración audiométrica en alumnos y personal del CUCBA.* Obtenido de http://repositorio.cucba.udg.mx:8080/xmlui/handle/123456789/5018

Lengua, I., Dunai, L., Peris-Fajarnés, G., & Brusola, F. (2013). Localización de sonidos en entorno abierto por personas ciegas. *Revista de Acústica*, (44) 25-29.

Lucio, J. (2021). *Material Clases Materia Simulación.* Quito.

Martínez R, E. B., Ramírez, N. C., Mesa H, G. A., Suárez, I. R., Trejo M, D. C., León, P. P., & Morales S, L. B. (2009). Árboles de decisión como herramienta en el diagnóstico médico. *Revista médica de la Universidad Veracruzana, 9(2)*, 19-24.

MATLAB. (2022). Obtenido de https://www.mathworks.com/products/matlab.html

Navlani, A. (2018). *Clasificación KNN utilizando Scikit-learn . Datacamp*. Obtenido de https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn.

Peralta, J., Carrión, L., Tenesaca, J., & Vázquez Rodas, A. (2017). *Clasificación de los sonidos cardíacos usando ondículas y redes neuronales* (Vol. 8). Cuenca, Ecuador: Maskana. Obtenido de https://publicaciones.ucuenca.edu.ec/ojs/index.php/maskana/article/view/2000

Recodo Estévez, S. (2017). *Implementación en tempo real dun sistema de auralización (Doctoral dissertation).*

Rey, D., & Gil, L. (2013). *Valoración subjetiva de las preferencias acústico-musicales del bambuco y pasillo colombiano (Ingeniería de sonido).* UNIVERSIDAD DE SAN BUENAVENTURA.

Russell, S., & Norvig, P. (2004). *Inteligencia artificial Un Enfoque Moderno (2nd ed.).* Madrid: Pearson Educación.

scipy.signal.correlate — Manual de SciPy v1.8.0 . Docs.scipy.org. (2022). Obtenido de https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.correlate.html.

Valencia Palma, A. (2019). *Localización del sonido utilizando beamforming y su representación en un dispositivo de realidad mixta.* Querétaro: Universidad Autónoma de Querétaro.

Vorländer, M., & Summers, J. E. (2008). *Auralization: Fundamentals of acoustics, modelling, simulation, algorithms, and acoustic virtual reality. Acoustical Society of America Journal, 123(6), 4028.* .

8. ANEXOS

ANEXO 1

SCRIPT createRays

```
function [outx,outy,outz] = createRays(NumberOfRays)
    A = [0 1 0; 0 2 0; 0 3 0; 0 4 0; 0 5 0;
        1 6 0; 2 6 0; 2 7 0; 3 7 0; 3 8 0;
        4 8 0; 4 9 0; 5 9 0; 5 10 0;1 10 0;
        6 11 0;7 11 0;8 11 0;9 11 0;10 11 0;
        1 2 0; 2 3 0; 3 4 0; 4 5 0; 5 1 0;
        6 7 0; 7 8 0; 8 9 0; 9 10 0;10 6 0
];
    T= [0 1 0;   1 2 0;   2 3 0;   3 4 0;   4 0 0;
        5 6 -1;  6 7 0;   7 8 -1;  8 9 0;   9 10 -1;
        10 11 0; 11 12 -1;12 13 0; 13 14 -1;14 5 0;
        15 16 -1;16 17 -1;17 18 -1;18 19 -1;19 15 -1
];
    %int i,j,k,n,m,RAY;
    %double S,R,xB,yB,zB,xC,yC,zC,c[8];
    %create Rays matrix
%       c=zeros(8,1);

    n=double(int64(floor(sqrt((NumberOfRays-2)/10)+0.5)));
    NRAYS=int64(2+10*(n^2));
    RaysX = zeros(NRAYS,1);
    RaysY = zeros(NRAYS,1);
    RaysZ = zeros(NRAYS,1);

    %calculating the icosaedron vertives
    S=2/sqrt(5);
```

```
R=(5-sqrt(5))/5;
RaysX(1)=0;
RaysY(1)=0;
RaysZ(1)=1;

for i=2:1:6
   RaysX(i)=S*cos(( pi*(i-1)*72)/180);
   RaysY(i)=S*sin(( pi*(i-1)*72)/180);
   RaysZ(i)=1-R;
   RaysX(i+5)=S*cos((72* pi*(i-1))/180+(36* pi)/180);
   RaysY(i+5)=S*sin((72* pi*(i-1))/180+(36* pi)/180);
   RaysZ(i+5)=R-1;
end
RaysX(12)=0;
RaysY(12)=0;
RaysZ(12)=-1;
RAY=13;
%calculating the rays on the icosaedron edges
for j=1:1:30
   A(j,3)=RAY;
   xB=RaysX(A(j,1)+1);
   yB=RaysY(A(j,1)+1);
   zB=RaysZ(A(j,1)+1);
   xC=RaysX(A(j,2)+1);
   yC=RaysY(A(j,2)+1);
   zC=RaysZ(A(j,2)+1);
   c(1)=(xC^2)*((yB^2)+(zB^2))+((yC*zB-yB*zC)^2)-2*xB*xC*(yB*yC+zB*zC)+(xB^2)*((yC^2)+(zC^2));
   c(2)=acos(xB*xC+yB*yC+zB*zC);
   c(3)=-xC*(yB*yC+zB*zC)+xB*((yC^2)+(zC^2));
   c(4)=xC*((yB^2)+(zB^2))-xB*(yB*yC+zB*zC);
   c(5)=(xC^2)*yB-xB*xC*yC+zC*(-yC*zB+yB*zC);
```

```
    c(6)=-xB*xC*yB+(xB^2)*yC+zB*(yC*zB-yB*zC);
    c(7)=(xC^2)*zB-xB*xC*zC+yC*(yC*zB-yB*zC);
    c(8)=-xB*xC*zB+(xB^2)*zC+yB*(-yC*zB+yB*zC);
    for i=1:1:n-1
        RaysX(RAY)=(c(3)*cos(double(i*c(2)/n))+c(4)*cos(double((n-i)*c(2)/n)))/c(1);
        RaysY(RAY)=(c(5)*cos(double(i*c(2)/n))+c(6)*cos(double((n-i)*c(2)/n)))/c(1);
        RaysZ(RAY)=(c(7)*cos(double(i*c(2)/n))+c(8)*cos(double((n-i)*c(2)/n)))/c(1);
        RAY=RAY+1;
    end
end
%calculating the rays on the icosaedron faces

for k=1:1:20
    for j=1:1:n-1
        xB=RaysX(A(T(k,1)+1,3)+j-1);
        yB=RaysY(A(T(k,1)+1,3)+j-1);
        zB=RaysZ(A(T(k,1)+1,3)+j-1);
        xC=RaysX(A(T(k,2)+1,3)+j-1);
        yC=RaysY(A(T(k,2)+1,3)+j-1);
        zC=RaysZ(A(T(k,2)+1,3)+j-1);
        c(1)=(xC^2)*((yB^2)+(zB^2))+((yC*zB-yB*zC)^2)-2*xB*xC*(yB*yC+zB*zC)+(xB^2)*((yC^2)+(zC^2));
        c(2)=acos(xB*xC+yB*yC+zB*zC);
        ppp=xB*xC+yB*yC+zB*zC;
        lll=acos(xB*xC+yB*yC+zB*zC);
        c(3)=-xC*(yB*yC+zB*zC)+xB*((yC^2)+(zC^2));
        rrrr=-xC*(yB*yC+zB*zC)+xB*((yC^2)+(zC^2));
        c(4)=xC*((yB^2)+(zB^2))-xB*(yB*yC+zB*zC);
        c(5)=(xC^2)*yB-xB*xC*yC+zC*(-yC*zB+yB*zC);
        c(6)=-xB*xC*yB+(xB^2)*yC+zB*(yC*zB-yB*zC);
        c(7)=(xC^2)*zB-xB*xC*zC+yC*(yC*zB-yB*zC);
        c(8)=-xB*xC*zB+(xB^2)*zC+yB*(-yC*zB+yB*zC);
```

```
            if T(k,3)==0
               m=j;
            else
               m=n-j;
            end
            for i=1:1:m-1
               RaysX(RAY)=(c(3)*cos(double(i*c(2)/m))+c(4)*cos(double((m-i)*c(2)/m)))/c(1);
               RaysY(RAY)=(c(5)*cos(double(i*c(2)/m))+c(6)*cos(double((m-i)*c(2)/m)))/c(1);
               RaysZ(RAY)=(c(7)*cos(double(i*c(2)/m))+c(8)*cos(double((m-i)*c(2)/m)))/c(1);
               RAY=RAY+1;
            end
         end
      end
      outx=RaysX;
      outy=RaysY;
      outz=RaysZ;
end
```