



Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

Detección de biomarcadores de Parkinson en la voz usando redes SOM

Autor: Jesús López Bermejo
(Ingeniería de Software)

Director: Francisco Díaz Pérez

2023

1. resumen

2. abstract

Índice de figuras

1.	Comparativa entre un cerebro sano y uno con la EP [17]	9
2.	Elementos del aparato fonador[27]	11
3.	Espectrograma y oscilograma de la palabra <i>acuso</i> [8]	12
4.	Señal microfónica con medidas en frecuencia de los ciclos vocales[10]	12
5.	Señal microfónica con medidas en amplitud de los ciclos vocales[10]	13
6.	Comparación entre un espectrograma de una persona sana y una persona con la EP[39]	15
7.	Asginación de pesos a las entradas y salida de una neurona[19]	17
8.	Representación visual de una red de neuronas multicapa[22]	17
9.	Selección de BMU en una red 5x5	18
10.	Distribución de los registros de una red de neuronas auto-organizadas (SOM) entrenada	19
11.	Distancias entre neuronas de una red de neuronas auto-organizadas (SOM) entrenada	19
12.	Librerías cargadas en la sesión actual	21
13.	Estadística del <i>Iris DataSet</i> de prueba	21
14.	Gráfico de número de veces elegida cada neurona como BMU	22
15.	Cantidad de veces elegida una neurona como BMU y grupos hechos por la red SOM en una red 3x3	22
16.	Cantidad de veces elegida una neurona como BMU y grupos hechos por la red SOM en una red 3x3	23
17.	Código para la obtención de la estadística	24
18.	Estadística del <i>DataSet</i>	24
19.	Código para la estadística del Multiple Sound <i>DataSet</i>	28
20.	Información sobre los datos del <i>DataSet</i>	28
21.	Representación de todas las características en diagramas de bigotes	32
22.	Representación de todas las características en diagramas de bigotes una vez aplicado el <i>StandardScaler</i>	33
23.	Representación de todas las características en diagramas de densidad solapados una vez aplicado el <i>StandardScaler</i>	33
24.	Matriz de correlación del <i>DataSet</i>	35
25.	Matriz de correlación del <i>DataSet</i> filtrada por valores con correlacion ≥ 0.6	36
26.	Puntuación de las características según el <i>Feature Selection</i>	36
27.	Puntuación de la <i>Mutual Information</i> del <i>DataSet</i>	37
28.	Matriz de correlación filtrada, por correlación mayor o igual a 0.6, con grupos de correlación	39
29.	Distancias entre las neuronas de las 4 redes formadas con distintas <i>seeds</i>	41
30.	Número de elecciones como BMU de cada neurona de la red	42
31.	Distancias entre las neuronas de la red con sus neuronas vecinas	42
32.	Árbol de relaciones jerárquicas o dendrograma de los <i>clusters</i>	43
33.	Grupos formados por la red para la imposición de 2 grupos	44
34.	Grupos formados por la red para la imposición de 3 grupos	44
35.	Grupos formados por la red para la imposición de 4 grupos	45
36.	Matriz con la distribución de personas enfermas	47
37.	Matriz con la distribución de personas sanas	47
38.	Diferencia entre la cantidad de sanos y enfermos por neurona (positivo y verde para sanos, negativo y rosa para enfermos)	48
39.	Pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos del experimento 1	49
40.	Diferencia entre la cantidad de sanos y enfermos por neurona	50
41.	Pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos del experimento 2	50
42.	Diferencia entre la cantidad de sanos y enfermos por neurona	51
43.	Pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos del experimento 3	51
44.	Diferencia entre la cantidad de sanos y enfermos por neurona	52

45.	Pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos del experimento 7	52
46.	Diferencia entre la cantidad de sanos y enfermos por neurona	53
47.	Pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos del experimento 8	53
48.	Conjunto de gráficas sobre los pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos por modelo	53

Índice de tablas

1.	Descripción de las características del Oxford Parkinson's Disease Detection <i>DataSet</i>	26
2.	Descripción de grupos para el <i>Oxford DataSet</i>	27
3.	Características del <i>DataSet</i> totales	29
4.	Descripción de grupos para el <i>Multiple Sounds DataSet</i> [39]	31
5.	Valores del KurtosisTest por característica	34
6.	Especificación del valor de N según el método de estandarización elegido	37
7.	Especificación del valor de E según el método de elección de características	38
8.	Grupos asignados para los registros de voz pertenecientes a 5 personas, una de ellas no padece la EP	46

Índice

1. resumen	1
2. abstract	2
3. Introducción	7
3.1. Motivación	8
3.2. Objetivos	8
4. La enfermedad de Parkinson y la voz	9
4.1. La enfermedad de Parkinson	9
4.2. La voz	11
4.2.1. Parametrización de la voz	12
4.3. Conexión entre la EP y la voz	15
5. Herramientas	15
5.1. Inteligencia Artificial	15
5.1.1. Clustering	16
5.2. Redes de Neuronas	16
5.3. Red de neuronas auto-organizadas (SOM)	18
5.4. Python y R	19
5.5. Librerías	20
5.5.1. Sklearn	20
5.6. Ejemplo de Red SOM en R	21
6. Exploración de <i>DataSets</i>	23
6.1. Oxford Parkinson's Disease Detection <i>DataSet</i>	24
6.2. Multiple Sounds UCI <i>DataSet</i>	27
6.3. Elección de <i>DataSet</i>	31
7. Procesamiento y modelado	32
7.1. Introducción	32
7.2. Exploración y análisis de los datos	32
7.3. Preprocesamiento	38
7.4. Construcción de los modelos	40
7.4.1. Experimento 1: G01_SC_22_10x10	40
7.4.2. Experimento 2: G02_SC_22_10x10	49
7.4.3. Experimento 3: G02_SC_22_10x10	50
7.4.4. Experimento 4: G04_SC_22_10x10	51
7.4.5. Experimento 5: G05_SC_22_10x10	51
7.4.6. Experimento 6: G06_SC_22_10x10	51
7.4.7. Experimento 7: G07_SC_22_8x8	52
7.4.8. Experimento 8: G08_SC_12_8x8	52
8. Estudio y análisis	54
9. Conclusiones	54
9.1. Implicaciones Sociales y Medioambientales	54
10. Bibliografía	55
11. Anexos	58

3. Introducción

La enfermedad de Parkinson, por sus siglas EP, es la patología neurodegenerativa motora con mayor incidencia a nivel mundial, teniendo una mayor aparición en adultos a partir de los 60 años[47]. Esta enfermedad afecta a 160.000 personas en España y a más de 7 millones de personas en todo el mundo[15], causando daños o incluso la muerte de las células nerviosas de nuestro cerebro que controlan el movimiento de todo el cuerpo, teniendo entre sus síntomas más prematuros los trastornos en la voz y en el habla de las personas. Las previsiones a futuro no mejoran, ya que para 2040 se esperan alrededor de 12 millones de afectados por la EP[14]. Además, los datos aumentan de relevancia cuando se destaca que los trastornos neurológicos son una de las principales causas de discapacidad en el mundo, y en concreto la EP es la que parece acusar la mayor tasa de crecimiento de todas las patologías neurodegenerativas[34].

Aunque han habido múltiples estudios e intentos de predecir la enfermedad de Parkinson a través de herramientas como la que se presentan en este trabajo, la detección de EP se sigue realizando por un diagnóstico clínico al uso[24]. Este diagnóstico implica que el médico vea suficientes razones para mandar al paciente hacer las pruebas, lo que puede conllevar mucho tiempo debido a los usuales retrasos en la realización de pruebas como en la Comunidad Valenciana entre el 2013 y 2017, donde se llegó a observar un retraso diagnóstico de unos 12 meses [44]. Dependiendo de los síntomas que observe el médico puede mandar distintas pruebas, como tomografía computarizada por emisión de fotón único[40]. Esta prueba consiste en utilizar sustancias radioactivas para crear una imagen tridimensional de como funciona un órgano, usándose para ver la cantidad de transportador de dopamina posee el sujeto[41]. Otras pruebas que suelen realizarse para diagnosticar la EP suelen ser para descartar que sea otra enfermedad o infecciones, como análisis de sangre, ecografías cerebrales, resonancias magnéticas o la tomografía por emisión de positrones[40]. Además, otra opción de diagnóstico es el sometimiento a ciertos medicamentos contra la EP, y si tiene un efecto positivo, es por que muy probablemente padezcas de la enfermedad de Parkinson[40]. La creación de herramientas de apoyo para el personal especializado en la detección de enfermedades es clave para la reducción de tiempo en dicha detección, ya que estas pruebas pueden tardar meses o incluso años en confirmar si se padece la EP o no[34]. Esto se debe a que se requiere de la realización de varias pruebas genéricas, ya que no existen actualmente pruebas concretas que puedan verificar si se padece la EP[40]. Aunque se están empezando a estudiar ciertas pruebas específicas para la identificación de la EP, aún no están aceptadas. Otro motivo de la tardanza de los resultados del diagnóstico es la comprobación de los resultados obtenidos, ya que a cada persona las pruebas realizadas pueden conllevar a resultados distintos, pese a padecer los mismos síntomas[24].

Hay una necesidad imperiante de conseguir reducir el tiempo del diagnóstico en el mínimo posible. Pero ya no solo eso, existe un problema generalizado con las grandes ciudades, y es que hay una tendencia mayoritaria de acumular recursos y servicios en estas grandes ciudades, por lo que deja con menos recursos para ciudades más pequeñas. Esto pasa también con los médicos especializados en neurología, haciendo que el ratio de neurólogos cada 100.000 personas, se reduzca, llegando a ser menor del aconsejado por la European Federation of Neurological Societies (EFNS) y la Sociedad Española de Neurología (SEN)[13]. Este ratio aconsejado es de 5 neurólogos cada 100.000 habitantes. En las grandes ciudades como Madrid sí se consigue un buen ratio, pero no es lo más común. Un ejemplo de este problema es Málaga, teniendo un ratio de 2,3 neurólogos cada 100.000 habitantes, siendo menos de la mitad de lo recomendado. Incluso en el global de España sigue siendo un ratio pobre, siendo este de 3 neurólogos cada 100.000 habitantes[20]. Estos datos nos hacen ver la mala repartición de los neurólogos, y la importancia de conseguir diagnósticos de calidad y en un tiempo mínimo, para poder atender a la mayor cantidad de personas posibles con el personal actual.

En este marco es en el que se desarrolla el proyecto. El tiempo juega un factor clave en el tratamiento de la enfermedad, ya que los tratamientos dependen en gran medida de la pronta diagnosis en un paciente, y como acabamos de ver, se requiere de mayor personal especializado, por lo que ayudaría tener una herramienta que agilice la detección de enfermedades neurodegenerativas, en el caso particular de este proyecto, la EP. Específicamente en este trabajo se procesa un conjunto de datos extraído de una base de datos publicada por la University of California Irvine (UCI), y se ha utilizado para entrenar el modelo, y para validarlo.

3.1. Motivación

El estudio de enfermedades siempre ha sido un campo en el que la innovación y la vanguardia ha sido clave para encontrar soluciones y mejoras en los modelos existentes, incluso la creación de nuevas formas de entender las enfermedades. No es muy distópico pensar que los avances en Inteligencia Artificial podrían aplicarse a todos los campos de la ciencia, y obtener resultados inimaginables. De hecho, existen estudios donde se comparan diagnósticos de médicos profesionales con los diagnósticos realizados por una Inteligencia Artificial, donde la IA obtuvo una mayor precisión que la mayoría de los médicos que participaron en dicho estudio[38]. Esto es una noticia fantástica por lo que implica. Todo lo que implique una mejora en la detección de enfermedades es bienvenida. Por ello, este tipo de investigaciones son necesarias ya que cuánto más se investigue, mejores resultados se acaban obteniendo, lo que significa que potencialmente se pueden salvar más vidas. Este proyecto nace de querer ayudar a la comunidad científica, más concretamente a la rama de la medicina, ya que a través de este proyecto, se podrán observar y analizar relaciones entre parámetros que pueden estar relacionados directamente con la evolución de la EP.

3.2. Objetivos

El objetivo principal de este proyecto se centra en la creación de varios modelos de redes de neuronas auto-organizadas (SOM), para después, seleccionar el modelo con una mejor separación en regiones de personas sanas y personas con la EP. Aunque el objetivo está claro, es necesario identificar objetivos menores para describir de forma mas precisa y exhaustiva el objetivo general. Estos pequeños objetivos son puntos clave dentro del proyecto, y la consecución de ellos conllevará a que, en su totalidad, se complete el objetivo en general. Los objetivos son los siguientes:

- Entender qué es la enfermedad del Parkinson.
- Entender la correlación entre la enfermedad del Parkinson y la voz.
- Explorar distintos *DataSets* sobre la representación de la voz de personas con la EP.
- Selección del mejor *DataSet* para el proyecto.
- Tratamiento de los datos del *DataSet*.
- Selección de características con *lass* que se crearán los modelos.
- Entrenamiento de los modelos.
- Selección del modelo más preciso para la distinción de personas sanas y personas con la EP.
- Realizar un análisis de los resultados, estudiando la utilidad del modelo en un ambiente real.

4. La enfermedad de Parkinson y la voz

En esta sección abordaremos la conexión existente entre la enfermedad de Parkinson (EP) y la voz, ya que no se entiende un trabajo sobre dicha conexión, sin explicarla. Primeramente se explicará que es la enfermedad de Parkinson, tras ello una breve definición sobre la voz y sus características, para acabar hablando de la conexión entre ambas.

4.1. La enfermedad de Parkinson

La enfermedad de Parkinson (EP) es la segunda enfermedad neurodegenerativa con mayor número de afectados, después del Alzheimer[15], y la primera con mayor tasa de crecimiento dentro de las enfermedades neurodegenerativas, siendo aún una enfermedad sin cura, aunque con tratamientos que ralentizan su progresión. La EP se produce por la pérdida de neuronas en una región del cerebro llamada *sustancia negra*, encargada de generar dopamina y de funciones como la regulación del movimiento voluntario[7]. La dopamina es un neurotransmisor que modula el sistema de los ganglios basales que controla el movimiento, del que si no se tiene un nivel adecuado, se generan los síntomas característicos de la EP. Actualmente no se conocen las causas de la enfermedad del Parkinson, aunque existen evidencias de algunos factores que favorecen la aparición de esta enfermedad, como son el envejecimiento, factores genéticos y exposición a toxinas como herbicidas[15].



Figura 1: Comparativa entre un cerebro sano y uno con la EP [17]

La imagen de la izquierda de la Figura 1 pertenece a un cerebro de una persona sana, mientras que el de la derecha corresponde a una persona con la EP, pudiendo verse claramente como desaparece la *sustancia negra* en los pacientes con la EP.

Respecto a los síntomas podemos encontrar entre otros los siguientes[16]:

- **Temblores**

Los temblores suelen empezar por una extremidad y acaban extendiéndose al resto de extremidades. Suele ser común que estando en reposo empiecen los temblores en alguna extremidad, y durante alguna actividad donde esté involucrada dicha extremidad no aparezcan los temblores.

- **Disartria hipocinética**

La disartria hipocinética es un tipo de disartria originada por lesiones en el Sistema Extrapiramidal[4]. La disartria es una alteración en el habla, sobretudo en su articulación debido a lesiones en el sistema nervioso central. En el caso de la disartria hipocinética se caracteriza por movimientos repetitivos causando monotonía en el habla, voz débil, movimientos lentos, articulación del habla defectuosa...[4] La mayoría de personas con la EP padecen de este síntoma (entre el 60-80 %), y aumenta progresivamente su gravedad conforme avanza la enfermedad[12].

- **Rigidez muscular**

Puede ocurrir en cualquier parte del cuerpo y conlleva dolor en los músculos y limitación en el rango de movimiento.

- **Bradicinesia**

Bradicinesia es la lentitud en los movimientos, sobretodo en los movimientos complejos, y también se caracteriza por producir incapacidad para iniciar y desarrollar movimientos con una velocidad normalizada. Esto produce que los movimientos de las personas que padecen bradicinesia tengan un movimiento menos vigoroso y menos intenso. Este síntoma es de los más característicos y visibles de la enfermedad de Parkinson[42].

También existen otros síntomas muy comunes como los problemas en la piel debido a alteraciones del sistema nervioso autónomo. Algunos de los más comunes suelen ser dermatitis seborreica y la hiperhidrosis, llegando a afectar al 30 % de los pacientes con la EP, incluso se llega a desarrollar un mayor riesgo a padecer melanomas[37].

Para diagnosticar la EP requiere que el paciente tenga Bradicinesia y al menos una de las siguientes señales[25]:

- **Rigidez muscular**

- **Temblor de reposo de 4-6 Hz**

- **Inestabilidad postural no causada por disfunción visual, vestibular, cerebelosa o propioceptiva**

Una problemática de la detección de la EP es que para poder observar estas señales, la enfermedad tiene que estar avanzada[11]. Para medir como de avanzada está la enfermedad se pueden usar diferentes escalas y medidas. A continuación se pasa a listar las escalas usadas[11] para medir la EP:

- **Escala Unificada de Valoración de la Enfermedad de Parkinson (UPDRS)**

Esta es la escala más usada y más fiable, ya que es una escala multidimensional y con pocos inconvenientes. En esta escala se miden el funcionamiento mental, la conducta, el ánimo, las actividades de la vida cotidiana y la función motora.

- **Escala de Hoehn y Yahr**

Permite realizar comparaciones entre poblaciones de pacientes con la EP pero no valora los cambios clínicos en un mismo paciente.

- **Escala de Ziegler-Bleton**

Es una escala de clasificación simple que evalúa la progresión de la discapacidad motora del paciente.

- **Escala de NUDS**

Esta escala evalúa la discapacidad producida por la enfermedad en las actividades de la vida diaria.

- **Escala de Webster**

Se caracteriza por evaluar rigidez, bradicinesia, postura parkinsoniana, balanceo de los brazos, marcha, temblor, fascies, habla, autocuidado y seborrea[6].

- **Escala de la Marcha para la Enfermedad de Parkinson**

Tiene cuatro apartados referentes a la Exploración, Capacidad funcional, Complicaciones a largo plazo y Ambito Social[33].

- **Escala de King College Hospital**

Esta escala es la más utilizada para medir el dolor físico de una persona con la EP. Consta de 14 preguntas, clasificadas según el tipo de dolor. Cada pregunta se contesta en dos apartados, intensidad y gravedad. Después se multiplican y suman las respuestas para obtener un valor entre 0 y 168[9].

■ Escala de ISAPD

La escala de ISAPD(Escala Intermedia de Valoración para la Enfermedad de Parkinson) consta de 13 preguntas sobre cuánto tarda o cuanto le cuesta realizar tareas habituales en su día a día como podría ser ducharse o sentarse en una silla.

Estas escalas permiten cuantificar la magnitud de la EP y permite a los médicos asignar los mejores tratamientos y medicamentos, según el valor en diferentes escalas.

4.2. La voz

La producción de la voz es resultado de las vibraciones de las cuerdas vocales que, dependiendo de la tensión a la que estén sometidas, genera un sonido más agudo o más grave que se amplía al pasar por los resonadores principales (cavidad bucal y cavidad nasal). El ser humano no posee ningún órgano que sea específicamente productor de sonidos, si no un conjunto de elementos especializados en otras tareas que al trabajar conjuntamente, generan la voz. Entre estos elementos se encuentran partes del aparato digestivo(boca), aparato respiratorio(pulmones y la laringe) y una serie de grupos musculares (diafragma y la musculatura abdominal), formándose el aparato fonador. La glotis se encuentra en la laringe y consta de las cuerdas vocales y una abertura entre ellas, que es por donde circula el aire. Este aire es forzado a pasar por esta abertura, llamada rima glótica, haciendo vibrar las cuerdas vocales, produciendo la voz[43].

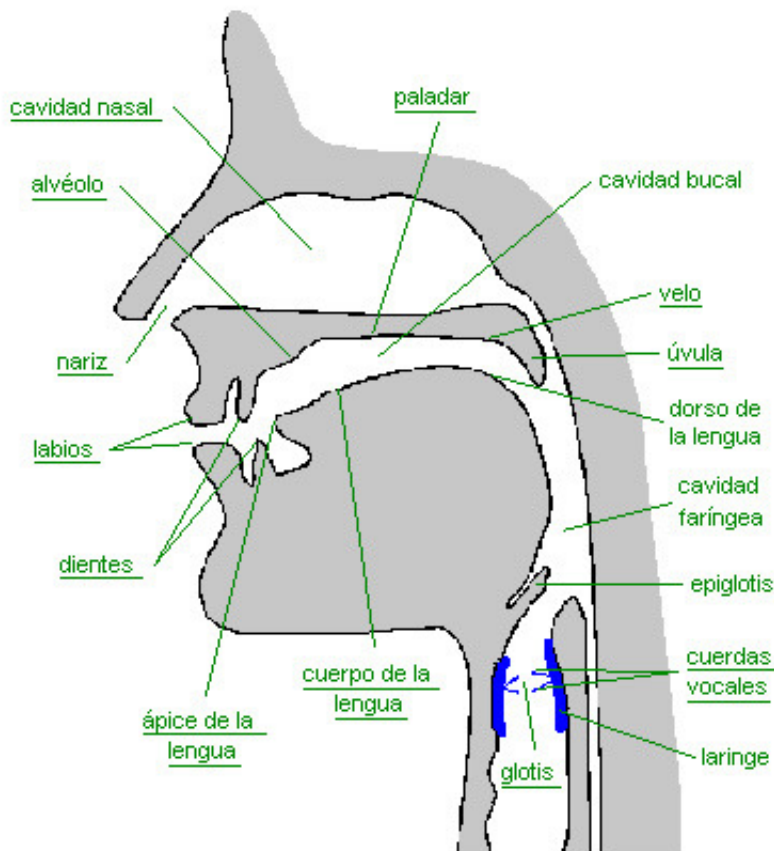


Figura 2: Elementos del aparato fonador[27]

La altura y la intensidad de la voz dependen de la estructura en conjunto de los órganos que componen el aparato fonador de cada individuo. Es por esto que la voz de un niño y de un adulto difieren, al igual que la de un hombre y una mujer o incluso de personas del mismo sexo[43].

4.2.1. Parametrización de la voz

La voz es una onda compleja que viaja a través del aire de la cual pueden extraerse diversos parámetros para ser el objeto de estudio en diversas áreas como la que acontece en este proyecto.

Una representación de la voz es el espectrograma. Un espectrograma es una representación de un sonido en el que se pueden diferenciar las distintas intensidades y amplitudes de la onda en el tiempo. Para ello, se utilizan transformadas de Fourier para descomponer esta onda compleja en un conjunto de seno y cosenos[23]. De esta forma se consigue descomponer una onda compleja en sus componentes espectrales individuales, pudiendo obtener información sobre la onda, en nuestro caso, la voz. Otra herramienta capaz de caracterizar un sonido es el oscilograma. Un oscilograma es la representación de la variación de presión de un sonido en el tiempo[21]. Los oscilogramas suelen ser usados para detectar variaciones de aire que crean compresiones y dilataciones en el tiempo, dando como resultado formas indeterminadas de fluctuaciones, comúnmente conocido como ruido[21].

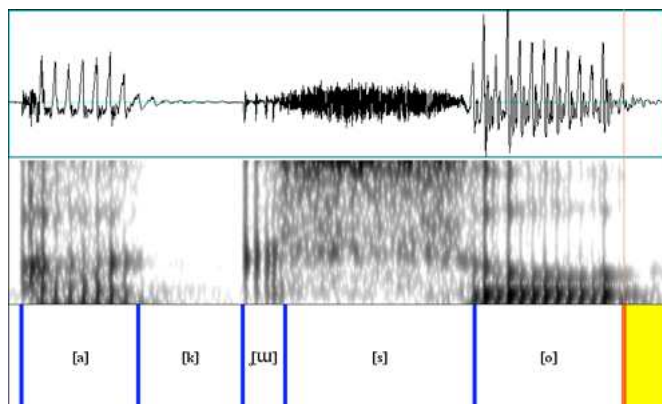


Figura 3: Espectrograma y oscilograma de la palabra *acuso*[8]

Después de este proceso de digitalización de la voz se extraen ciertas características específicas que serán relevantes para el estudio de la voz. Principalmente se pueden dividirse en parámetros que caracterizan la fuente de excitación vocal y parámetros que estudian el tracto vocal. El aumento de las perturbaciones de la frecuencia, de la intensidad y la presencia de ruido espectral son las principales alteraciones causantes de las voces disfónicas. Estas características son objeto de estudio a través de los siguientes parámetros extraíbles de la voz[10]:

■ Jitter

El *Jitter* es la perturbación de la variabilidad de la frecuencia ciclo a ciclo. Este parámetro representa una medida de la estabilidad de la fonación, siendo el conjunto de variaciones involutarias de la F0 que suceden de un ciclo a otro. Esta F0 es la frecuencia armónica, siendo el componente frecuencial más bajo de la señal microfónica. La F0 representa el número de veces que se abren y cierran las cuerdas vocales por segundo, expresado en Hz. Aunque la laringe humana es capaz de producir un rango de frecuencias, los valores normales se establecen en unos 125Hz para el hombre, 250Hz para la mujer y unos 350Hz para niños y niñas[10].

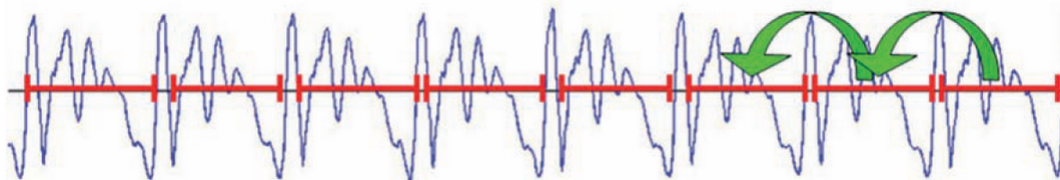


Figura 4: Señal microfónica con medidas en frecuencia de los ciclos vocales[10]

Como vemos en la Figura 4, cada línea roja representa un periodo, y las flechas verdes señalan distintos ciclos vocales. Vemos que aunque se parezcan mucho, nunca llegan a ser iguales, ya que el ciclo vocal responde a una vibración caótica, que aunque se intenten imitar las condiciones que se dan en los ciclos, resulta imposible que se den exactamente las mismas. El *Jitter* representa las variaciones de estas líneas rojas (en porcentaje de ciclos). Aunque parece que el *Jitter* es un claro indicador de disfonías, no sirve para determinar la causa de las disfonías, ya que hay multitud de variables que afectan a esta medida, como lesiones en las cuerdas vocales, ronquera e incluso la edad es un factor que cambia la medida del *Jitter*. Dependiendo del algoritmo que se aplique, puede obtenerse distintos tipos de *Jitter*[10]:

- *Jitter absoluto*

Mide la variación en tiempo de los ciclos totales, sin contar los primeros ni los últimos, ya que suelen ser zonas con una mayor inestabilidad vocal. Esta medida tiene una clara relación con el F0 de la persona, por lo que varía con el sexo.

- *Jitter relativo*

Esta medida viene a representar lo mismo que el *Jitter absoluto*, pero en porcentaje. Esto implica que la F0 no afecte a la medida, gracias a utilizar como medida un porcentaje.

- *Jitter RAP*

En este caso, no se comparan ciclos con ciclos, si no grupos de tres ciclos con grupos de tres ciclos, consiguiendo no tener en cuenta la variación de tono que se suele producir de forma periódica. Al igual que el *Jitter relativo*, se mide en porcentaje por lo que no afecta la distinción de sexo. Las iniciales RAP provienen del inglés *Relative Average Perturbation*.

- *Jitter PPQ*

El *Jitter PPQ* proveniente del inglés *Pitch Period Perturbation Quotient* y mide los cambios de la F0 en grupos de 5 en 5 ciclos, lo que implica que las diferencias entre los ciclos se suavizan, pudiendo estudiar la onda más uniformemente.

■ Shimmer

El *Shimmer*, al igual que el *Jitter* medía la perturbación en la frecuencia, mide la perturbación en la variabilidad de la amplitud, que representa la intensidad del ciclo. Esta intensidad es la amplitud de la señal microfónica en cada ciclo, que se forma al variar la presión sonora producida por la voz, y se expresa en decibelios(dB) y normalmente se encuentra entre 75 y 80 dB. Esta medida depende fundamentalmente de la amplitud de la vibración de las cuerdas vocales y de la presión subglótica. Existen algunos factores que producen una disminución en esta intensidad en la voz, como unas cuerdas vocales poco flexibles o un soporte respiratorio inadecuado. Esta intensidad decrece con el cuadrado de la distancia, por lo que siempre que se haga referencia a este parámetro, tiene que tenerse en cuenta la distancia a la que se ha grabado el registro de voz. Al igual que el *Jitter*, también representa una medida de la estabilidad de la fonación, aunque unas variaciones excesivas puede conllevar a padecer alguna patología, aunque al igual que el *Jitter*, no se puede vincular a ninguna patología específica. El *Shimmer* se relaciona de forma inversa con la intensidad vocal, disminuyendo su valor al aumentar la intensidad[10].

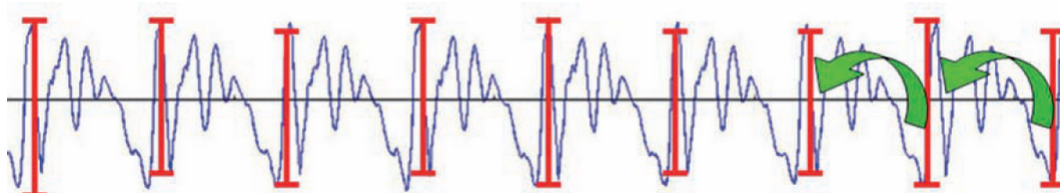


Figura 5: Señal microfónica con medidas en amplitud de los ciclos vocales[10]

Al igual que anteriormente, en esta Figura 5 aparecen líneas rojas, representando esta vez la amplitud de cada ciclo vocal, y las flechas verdes representan los cambios entre ciclos vocales, pudiendo tener

una amplitud mayor o menor. Paralelamente al *Jitter*, dependiendo del algoritmo que se utilice en la extracción del *Shimmer*, pueden encontrarse distintos tipos de *Shimmer*[10]:

- *Shimmer absoluto*

El *Shimmer* no varía dependiendo del sexo ya que no existen diferencias de intensidad en las señales microfónicas entre hombres y mujeres, pero sí que depende mucho del individuo, haciendo que los registros normales tengan un rango muy amplio.

- *Shimmer relativo*

Al igual que el *Jitter relativo* se mide en porcentaje, y compara ciclo a ciclo en este caso la amplitud.

- *Shimmer APQ y sAPQ*

Estos dos parametros vienen a representar lo mismo, que son las diferencias entre grupos de 5 en 5 pulsos glóticos y de 55 en 55, respectivamente.

- Parámetros de ruido

El ruido espectral se muestra como puntos entre los armónicos. En las voces normales hay escaso ruido interarmónico, lo que refleja un buen cierre de la glotis, mientras que las voces disfónicas poseen mucho ruido entre las líneas de los armónicos, y que estas propias líneas estén mal definidas. Por lo general cuanto más disfónica se la voz, más ruido tendrá su espectrograma[10].

- HNR medio

El *HNR medio* (Harmonic to Noise Ratio) mide la relación entre el componente que sucede periódicamente (armónicos) y los componentes que aparecen de forma inesperada y sin ningún patrón definido (ruido). Este parámetro compara dichos componentes de forma separada, como si se tratasen de dos ondas distintas, para posteriormente comparar sus intensidades.

- NNE medio

El *NNE* (Normaliced Noise Energy) transforma en una distribución normal los valores de la intensidad del ruido encontrado en la muestra analizada. Presenta valores negativos, y los más cercanos a cero son los más representativos para detectar patologías.

- NHR

El *NHR* (Noise to Harmonic Ratio) es la media del cociente de la energía inarmónica (entre 1500 y 4500Hz) y la energía armónica (entre 70 y 4500Hz). Esta energía inarmónica es la energía sonora no distribuida a través del espectro de manera ordenada matemáticamente. Normalmente se representa en relación con un fundamental (f)[26], por lo que el *NHR* se expresa como un porcentaje.

- Filtrado inverso

El filtrado inverso es una técnica no invasiva que refleja el movimiento vibratorio de las cuerdas vocales, recreando el movimiento de la glotis a través de un filtro que revierte la influencia del tracto vocal sobre ella. Esta técnica incluye varios parámetros[10]:

- Cociente de apertura (Open Quotient)

Es la relación entre el tiempo que permanece la glotis abierta y el periodo.

- Cociente de velocidad (Speed Quotient)

Es la relación entre el tiempo de apertura y el tiempo de cierre de la glotis.

- Cociente de cierre (Closing Quotient)

Es el porcentaje del tiempo que permanece cerrada la glotis.

4.3. Conexión entre la EP y la voz

Como hemos visto anteriormente, uno de los primeros síntomas de la EP tiene una estrecha relación con el habla, interviniendo en su deformación. El habla de una persona con la EP se suele caracterizar por tener una sonoridad e intensidad monótona que tiende a desaparecer al final de la fonación, por lo que dado un registro de voz con esta característica, tiene un alto porcentaje de que pertenezca a una persona con dicha enfermedad[18]. En ocasiones suele ocurrir que se suelen repetir las mismas sílabas, palabras o incluso frases completas, afectando a la frecuencia media del registro de voz[5]. Como vemos, hay múltiples motivos para asociar la EP con las características de la voz, por lo que parece razonable crear modelos que puedan identificar si una persona padece esta enfermedad o no en base a registros de voz grabados por diferentes personas. Además, la disartria hipocinética aparece en las personas con la EP con una frecuencia de entre el 60 % y el 80 %[12] y en tempranas edades de la enfermedad, por lo que los síntomas se aprecian desde el principio en la voz, siendo de gran utilidad para detectar la EP a través de la voz.

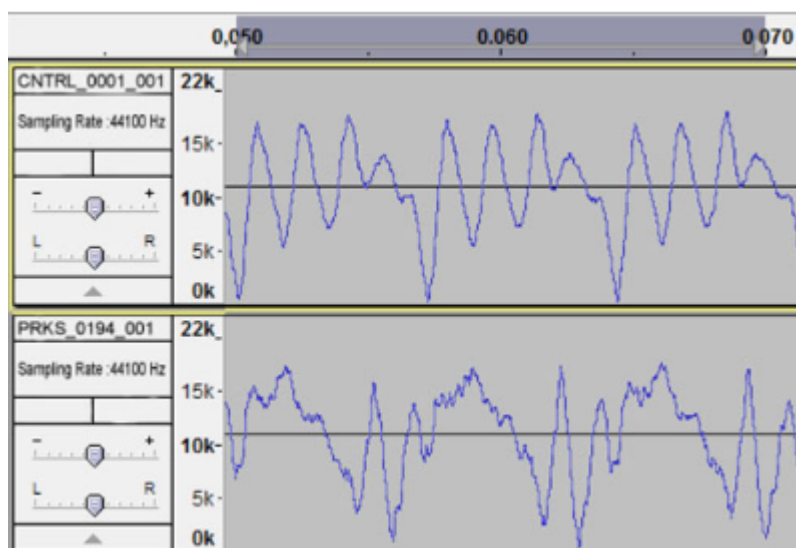


Figura 6: Comparación entre un espectrograma de una persona sana y una persona con la EP[39]

Como vemos en la Figura 6, una persona con la EP posee un espectrograma mucho más "débil", con muchas vibraciones durante todo el tiempo del espectrograma (ruido), mientras que una persona sana se mantiene con menos perturbaciones durante el tiempo

Hablar del resto de trabajos que tienen que ver con esto. Expandirme un poco

5. Herramientas

En esta sección se enumeran y definen las herramientas que se utilizan en el proyecto, desde lenguajes de programación, como tecnologías usadas. Se expondrán de forma más general, partiendo de las tecnologías en las que se basa el proyecto, pasando por los lenguajes de programación, hasta llegar a una mayor profundidad, entrando en las librerías usadas para el proyecto.

5.1. Inteligencia Artificial

La Inteligencia Artificial es la tecnología basada en replicar la inteligencia humana para realizar tareas. Esta Inteligencia Artificial es capaz de mejorar iterativamente a partir de la información que recogen o la información que se le da a dicha Inteligencia Artificial. Estas tareas se realizan de la misma forma

que lo harían los humanos, primero interpretan los datos externos, aprenden de dichos datos y emplean dichos conocimientos para realizar la tarea[1]. Los orígenes de esta idea se remontan a los años 30, con el considerado padre de la Inteligencia Artificial, Alan Turing, donde unas décadas más tardes planteaban la siguiente pregunta, "¿Pueden las máquinas pensar?"[1]. A partir de esta idea, surgieron muchos avances e intentos de averiguar si esto era cierto, y con ello, muchas ramas de la Inteligencia Artificial. Una de las áreas más importantes de la Inteligencia Artificial es el Machine Learning. Esta parte de la Inteligencia Artificial consiste en intentar conseguir que las máquinas aprendan a realizar tareas humanas, incluso predecir comportamientos[2]. Para ello, se necesitan cantidades de datos muy grandes, por eso su auge a la misma vez que se empiezan a conseguir cantidades de datos enormes gracias a la globalización de Internet. Los tipos de Machine Learning que existen se distinguen en tres grupos principales[2]:

- **Aprendizaje supervisado**

Se dotan a los modelos de los resultados esperados, y a partir de esa información el algoritmo aprende para generalizar esa información a cualquier dato parecido. Esto sirve para predecir resultados de lo que sea, ya sea resultados de partidos de fútbol, o si una persona es más frecuente a tener accidentes de tráfico y subirle el precio del seguro de coche. Dentro de esta categoría, se encuentran dos tipos de predicciones, clasificación y regresión. Clasificación se define como predicción a unos valores concretos o *Categorías*, de ahí su nombre, ya que clasifican los datos en categorías. Un ejemplo podría ser la clasificación de tipos de flores Iris a partir de su tamaño de pétalos. Por otra parte la regresión calcula datos continuos, como por ejemplo la predicción de la altura de una persona de ciertas características[2].

- **Aprendizaje no supervisado**

En este caso, el modelo tiene que aprender por su cuenta, sin tener la información sobre el resultado esperado. En este tipo de entrenamiento, el modelo directamente procesa los datos de cierta forma (dependiendo del algoritmo) para concluir con unas predicciones que pueden ser acertadas o no. El tipo de aprendizaje no supervisado principal es el *Clustering*, sobre el que se centra este proyecto, y que se explicará más a fondo más adelante[2].

- **Aprendizaje por refuerzo** Este tipo de Machine Learning es una mezcla de los dos anteriores, ya que antes se daba la información o no, pero en este tipo se hace ambas. Esto es posible a través de premios y castigos al modelo, sin darle directamente los resultados esperados. Cada vez que el modelo intenta predecir un resultado y esta mal, se le castiga señalándole que está mal el resultado, y el algoritmo por dentro cambia ciertos valores para aprender. En el caso contrario, cuando acierta un resultado se le señala como que es correcto, y el modelo por dentro refuerza los valores que tiene[2].

5.1.1. Clustering

El *Clustering* consiste en agrupar conjuntos de datos sin etiqueta con el fin de crear grupos o *Clusters*. Los datos de un *Cluster* son parecidos entre sí en términos de análisis, pero suficientemente diferentes con los datos de otros *Clusters*[32]. Esta segmentación en diferentes grupos se puede dar por características familiares y que corresponden a grupos ya detectados con anterioridad, o puede que el modelo agrupe en diferentes características ni siquiera valoradas por los humanos para hacer grupos[32]. En este proyecto se usará dicha tecnología para crear grupos de datos, intentando conseguir un modelo que separe correctamente entre personas sanas y personas con la EP.

5.2. Redes de Neuronas

Este trabajo propone como herramienta un modelo basado en Inteligencia Artificial que agrupa las características de los registros de voz en *Paciente con EP* o *Paciente sin EP*, más concretamente a través

de las redes de neuronas. Las redes de neuronas es un intento de implementación del funcionamiento de un cerebro humano, y cómo se interconectan las neuronas para obtener resultados a partir de una entrada. Una sola neurona es una función aplicada a unos valores de entrada, para generar una salida. Estos valores de entrada no siempre tienen la misma importancia en las neuronas, por eso cada neurona tiene pesos para las entradas. Estos pesos determinan cuánta importancia se les otorga a ciertas entradas. Durante el entrenamiento según pasa el tiempo, las neuronas aprenden más lento, es decir, el cambio en los pesos cada vez es menor.

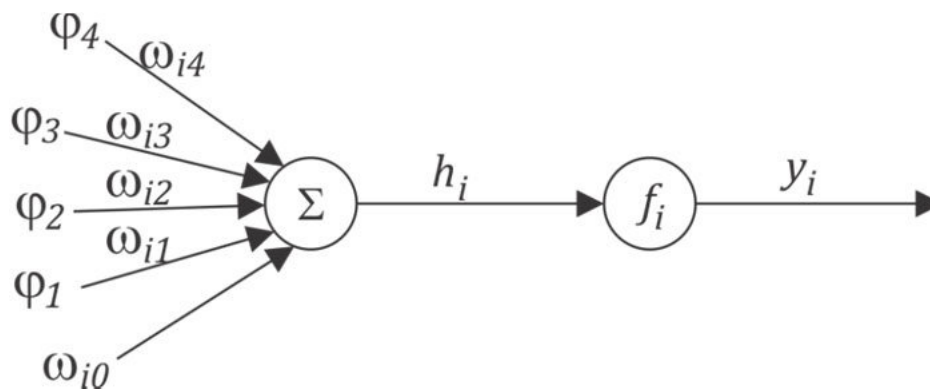


Figura 7: Asignación de pesos a las entradas y salida de una neurona[19]

En la Figura 7 las entradas están denotadas por la letra griega $\Phi(\phi)$, y los pesos por la letra griega $\Omega(\omega)$. Como vemos, cada peso está relacionado con cada una de las entradas, salvo uno comúnmente llamado *bias*, ya que se suele utilizar un peso suelto para inicializar la neurona, a parte de inicializar los pesos de las características. Estos pesos se multiplican a los valores de las entradas para dar más valor a ciertas entradas que la neurona considera más importantes. Estos pesos van modificándose según avanza el entrenamiento de la neurona, a través de funciones de realimentación, como por ejemplo el *Backpropagation*. Seguidamente se realiza el sumatorio de ellas y ese valor es el que entra a la función de la neurona (distinta dependiendo del modelo) llamada *función de activación*, y lo convierte en una salida. Si creamos varias neuronas que repitan este modelo para las mismas entradas entonces estaríamos formando una capa de neuronas y si las salidas de esta capa se pasan como entrada a otra capa de neuronas, estaríamos creando una red de neuronas multicapa.

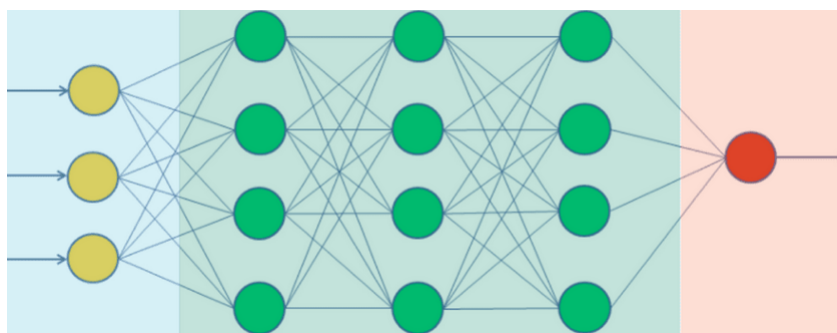


Figura 8: Representación visual de una red de neuronas multicapa[22]

En esta Figura 8, la primera columna de círculos representa la capa de entrada de la red de neuronas. Esta capa es la más superficial, recibiendo los datos y aplicando los pesos que irán variando según vaya pasando el tiempo. Las siguientes tres columnas de círculos verdes representan tres capas ocultas de neuronas. Estas capas están interconectadas entre ellas a cada una de las salidas de la capa anterior, y se encargan de procesar la información de la entrada. Finalmente vemos a la derecha un único círculo, que representa la capa de salida de la red de neuronas, que produce una conclusión. Otro concepto importante a conocer sobre las redes de

neuronas son los *Epochs*. Los *Epochs* o ciclos, son la cantidad de iteraciones que realizará el entrenamiento sobre todos los datos con los que se entrena. Normalmente se tiende a pensar que una gran cantidad de *Epochs* conllevan un mejor entrenamiento de la red, pero esto no suele ser así, ya que cuando se entrena una red de neuronas se pretende crear una red capaz de generalizar las características de los datos. En el caso de una cantidad demasiado grande de *Epochs* implica que la red se especialice en exceso, no logrando el objetivo deseado.

Partiendo de esta idea simple, se han implementado diversos tipos de redes de neuronas, desde redes más básicas como el Perceptrón hasta el Self-Organizing Map (SOM), que es el tipo que se usará en este trabajo.

5.3. Red de neuronas auto-organizadas (SOM)

Una red de neuronas SOM intenta recrear las zonas geográficas de la corteza cerebral. En la corteza cerebral no todas las regiones están especializadas en todas las funcionalidades del ser humano, si no que existen regiones de la corteza cerebral dedicadas a ciertas funcionalidades. Este es el comportamiento que imita la red de neuronas SOM. Esto se consigue premiando a la neurona que tenga la información más próxima a la entrada que se le pasa a la red de neuronas. Esta neurona elegida se le pasará a llamar Best Matching Unit (BMU)[28].

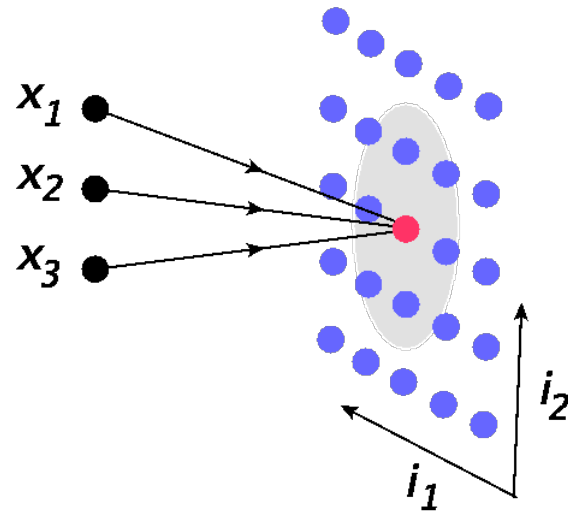


Figura 9: Selección de BMU en una red 5x5

En la Figura 9 se aprecia cómo la neurona central es elegida como la BMU, y a su alrededor se sombrea un área. Este área se conoce como *Vecindario*, y a todas las neuronas que se encuentren dentro de ese radio, también se les premiará, es decir se les otorgará un mayor valor en sus pesos durante el entrenamiento. Al igual que los pesos se van modificando con el tiempo, este área también, pasando a ser cada vez más pequeña, hasta que llegue un punto que sólo premia a la BMU. Los premios a las neuronas dentro del *Vecindario* también se irán reduciendo conforme vaya pasando el tiempo. Esta reducción del aprendizaje se llama ratio de aprendizaje. La proximidad a la información de la entrada viene determinado por los pesos de la neurona y los valores de la entrada. Las neuronas de una red de neuronas poseen un vector de pesos, normalmente notados con la letra $\Omega(\omega)$, siendo el tamaño de este vector exactamente la longitud de las características de las entradas. Todos estos valores serán inicializados aleatoriamente, ya que promedian mejores resultados que inicializándolos a un valor predeterminado. En cuanto a los valores de reducción del tamaño del radio del *Vecindario* y el ratio de aprendizaje, se utilizan valores que decrecen de forma exponencial, de tal forma que al principio, al no estar bien formado el mapa, necesitan aprender rápidamente y con un área suficientemente grande como para ir “especializándose” según van aumentando las iteraciones del algoritmo[28]. Al final del aprendizaje la red SOM quedará de la siguiente forma.

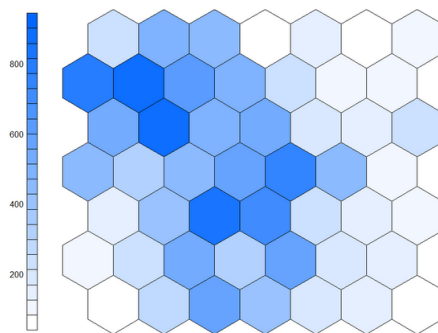


Figura 10: Distribución de los registros de una red de neuronas auto-organizadas (SOM) entrenada

En esta Figura 10 se observan hexagonos representando una neurona del modelo cada uno de ellos, con distintos grados de colores, que representan la cantidad de veces que esa neurona ha sido elegida como BMU. En la leyenda de la izquierda vemos que cuantas más ocasiones ha sido elegida la neurona como BMU, el azul tiene un tono más oscuro. Gracias a la representación visual que nos ofrece una red de neuronas auto-organizadas (SOM) se pueden dividir los datos en grupos. En la Figura 10 vemos que existe una región donde se eligen a las neuronas como BMU con una mayor frecuencia, pudiendo extraer esa zona del mapa como un grupo.

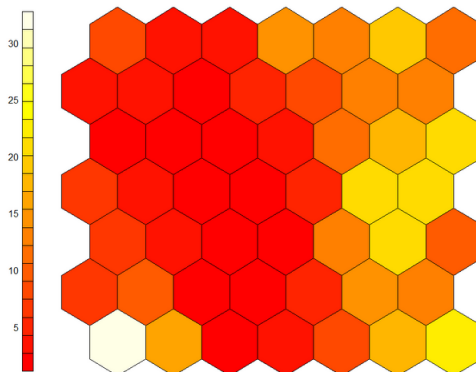


Figura 11: Distancias entre neuronas de una red de neuronas auto-organizadas (SOM) entrenada

En esta Figura 11 aparecen representadas las distancias entre las neuronas, es decir, como de parecidos se encuentran sus pesos. Esta imagen ayuda a la detección de grupos al igual que la Figura 11. Como especifica la leyenda de la izquierda, el color rojo intenso significa que la distancia entre una neurona y sus vecinas (neuronas colindantes) es pequeña, y si el color tiende a ser más claro, significa que la neurona se encuentra a una amplia distancia de las neuronas de su frontera.

5.4. Python y R

Como lenguajes de programación para llevar a cabo este proyecto se han elegido dos: Python y R. Ambos lenguajes se han elegido debido principalmente a la alta cantidad de herramientas con funcionalidades adecuadas para el proyecto. Es necesario el uso de estos dos lenguajes ya que en cada uno de ellos realizarán funciones distintas. En Python se desarrollará todo el código relativo a la preparación de los datos y el estudio de los mismos, a través de librerías populares como *Sklearn*, más desarrolladas en una sección posterior. En cuanto a R, se ha elegido por la gran cantidad de herramientas gráficas para la representación de redes

SOM. Ambos lenguajes son elegidos por la comodidad de trabajo con ellos, a parte de la experiencia previa con, sobretodo, uno de ellos, Python. Estos lenguajes son comúnmente utilizados por la comunidad científica para el desarrollo de tareas de *Machine Learning*, *Minería de Datos*, investigación bioinformática... En R se llevará a cabo el modelado y el entrenamiento de las redes SOM, y su posterior análisis, gracias a las herramientas de creación de redes SOM y herramientas gráficas que nos proporciona. Esta es una razón de peso para elegir R, la amplia serie de librerías ofrecidas para la creación de redes SOM.

5.5. Librerías

Una librería en un lenguaje de programación son un conjunto de funcionalidades probadas que facilitan la reutilización de código y la programación de código complejo. Al crear varios modelos de redes de neuronas SOM para este proyecto, necesitaremos utilizar librerías que nos permitan crear las redes, entrenarlas y representarlas gráficamente de diferentes maneras. La librería principal del proyecto es *Kohonen*[45][46], encargada de crear, y entrenar las redes SOM. También ofrece ciertas características de estas redes, como la posibilidad de pasar a tablas la cantidad de veces elegidas las neuronas como *BMUs*. Otra característica para elegir esta librería es la funcionalidad de dibujar distintos gráficos sobre la red. Esta librería posee otras características a las mencionadas, ya que permite expandir los mapas de la red SOM, crear mapas de calor para las distancias entre las neuronas, predicciones sobre datos de test, *DataSets* de prueba para configurar una red SOM, e incluso tiene unas redes llamadas *SuperSOM*. Estas redes son redes SOM de varias capas de redes de neuronas, a las que se les inyectan datos no bidimensionales como entrada. Esto quiere decir que son redes de neuronas SOM capaces de trabajar con matrices de matrices de datos. Para trabajar con ciertas estructuras de datos y poder eliminar columnas para guardarlas en otras variables fácilmente, utilizaremos de forma auxiliar *dplyr*, y *aweSOM* se usará para crear estadísticas del *DataSet* y para poder inicializar la red a nuestro gusto, respectivamente. Estas librerías sirven de apoyo para la inserción de los datos a la red SOM con el formato correcto y a una selección mayor de parametrización sobre la red de neuronas SOM, ya que la librería *Kohonen* no permite tantos parámetros de configuración, como por ejemplo la inicialización de las neuronas de la red. El preprocesamiento del *DataSet* se realizará en Python, por lo que se utilizan diversas librerías que permiten dibujar distintos tipos de gráficos, procesamiento de datos y creación de las propias redes SOM. Entre estas se encuentran Pandas, para obtener los datos del fichero y convertirlos en una estructura de datos llamada *DataFrame*, Numpy como librería auxiliar a Pandas para las estructuras de datos y obtener *Outliers* y rangos intercuartiles, Seaborn para dibujar principalmente gráficas y la Matriz de confusión, Matplotlib (más en concreto Pyplot) y OS (librería de Python para el intercambio de información con el sistema operativo) para guardar las imágenes y gráficas en archivos y directorios, y la más importante de este apartado, Sklearn, encargada principalmente del procesamiento de los datos.

5.5.1. Sklearn

Sklearn [36] es una librería Open Source para el desarrollo de machine learning dotando de herramientas para entrenamiento supervisado y no supervisado. También incluye funcionalidades para la selección de modelos, preprocesamiento de datos, evaluación de modelos y mucho más. Usaremos Sklearn principalmente para el preprocesamiento de los datos, utilizando distintas funciones que pasaremos a explicar sobre la estandarización de datos. Entre otros métodos de estandarización de datos, veremos principalmente cuatro.

- **MinMax Scaler**

Escala los valores al rango establecido. Este rango si no se establece con anterioridad se designa en $[0,1]$, siendo el 1 el máximo valor del conjunto de datos en valor absoluto. Este método tiene como principales ventajas la robustez a desviaciones típicas pequeñas y la preservación de entradas con valores de 0 en datos escasos.

- **Robust Scaler**

Utiliza la estadística para conseguir un método robusto a los *Outliers*. Este método borra la mediana y escala los datos a partir del rango intercuartil. Este rango intercuartil es el rango entre el primer cuartil y el tercero. Este método se suele utilizar cuando los *Outliers* tienen mucho peso negativo para el entrenamiento del modelo.

■ Standard Scaler

Estandariza los valores gracias a borrar la media e imponerla al valor elegido, junto a la desviación típica elegida anteriormente. Si no se determinan estos valores, por defecto son Media = 0 y Desviación Típica = 1. Existen muchos algoritmos que asumen dicha distribución para realizar el aprendizaje de los modelos, lo que permite que este modelo sea mejor que otros. A su vez, si las características tienen varianzas distintas, puede que se conviertan en características que dominen y que entrene mal la red.

■ MaxAbs Scaler

Se suele utilizar con datos escasos ya que mantiene las entradas con valores de 0. Es muy parecido a *MinMax Scaler*, pero en este método no se puede designar el rango, este directamente se obtiene a través del valor máximo del conjunto de datos en valor absoluto. El rango es siempre [0,1].

5.6. Ejemplo de Red SOM en R

Para dar más contexto del potencial de estos lenguajes en este área, vamos a ver un ejemplo en R del procedimiento que debemos seguir para crear un modelo de una red SOM, y los resultados de la misma. Primeramente necesitamos un *DataSet* de prueba para crear el modelo. Para ello, utilizaremos el *Iris DataSet*, utilizado comunmente para realizar pruebas de modelos y usado frecuentemente con carácter didáctico. Para ello importaremos las librerías necesarias, principalmente la librería *Kohonen* que nos permitirá crear la red SOM y entrenarla. Para cargar los datos y manipularlos utilizaremos *dplyr*, la cual nos permitirá crear *DataFrames*, una estructura de datos comúnmente utilizada para guardar *DataSets*, ya que nos permite realizar escalados, eliminación de columnas y renombrar columnas de forma sencilla. Si no están instalados, tendremos que instalarlos, y una vez instalados, cargarlos en la sesión actual. Cada vez que cerramos el IDE de R o cerramos la sesión, tendremos que cargar todas las librerías que utilicemos, ya que R utiliza sesiones y no se guardan las librerías cargadas. Una vez cargadas, llamando a la función *sessionInfo()* podremos ver si están cargadas correctamente las librerías.

```
other attached packages:
[1] plot3Drgl_1.0.4 rgl_1.0.1          plot3D_1.4      kohonen_3.0.11
[5] dplyr_1.1.0
```

Figura 12: Librerías cargadas en la sesión actual

Una vez instaladas y cargadas las librerías, pasamos a traer el .CSV con los datos, quitar la clase objetivo al ser un entrenamiento no supervisado y normalizar el *DataSet*. Esto se hace con el objetivo de igualar las escalas de las características. Si escribimos el comando *summary(df)*, podemos observar las estadísticas de los datos guardados en la variable *df*.

SepalLength	Sepalwidth	PetalLength	Petalwidth
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	1st Qu.:0.300
Median :5.800	Median :3.000	Median :4.400	Median :1.300
Mean :5.848	Mean :3.051	Mean :3.774	Mean :1.205
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500

Figura 13: Estadística del *Iris DataSet* de prueba

Siguiendo con el ejemplo, vamos a entrenar una red de neuronas de 7x7 hexagonal con un número de actualizaciones de 1000. Esta es una tarea sencilla gracias al uso de la librería *kohonen*, ya que con una

simple línea podemos llevar acabo el entrenamiento de toda la red de neuronas SOM. Una vez entrenada vamos a representar gráficamente el resultado.

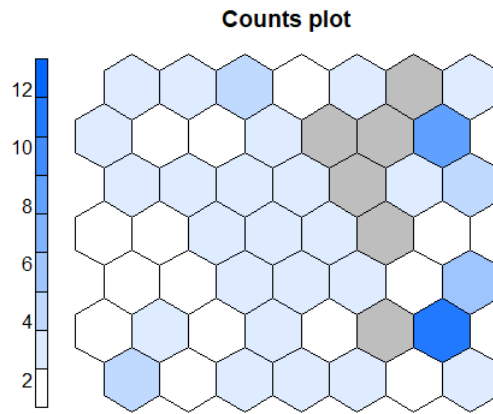


Figura 14: Gráfico de número de veces elegida cada neurona como BMU

Como observamos en la Figura 14, durante el entrenamiento han habido neuronas que nunca han sido elegidas como BMU, pudiendo indicar que el tamaño de la red es un poco grande para los datos existentes, mientras que han habido otras neuronas con una cantidad mucho mayor de elecciones como BMU. Esto se puede ver de forma más clara representando con puntos cada BMU elegida, junto a que *Cluster* pertenece.

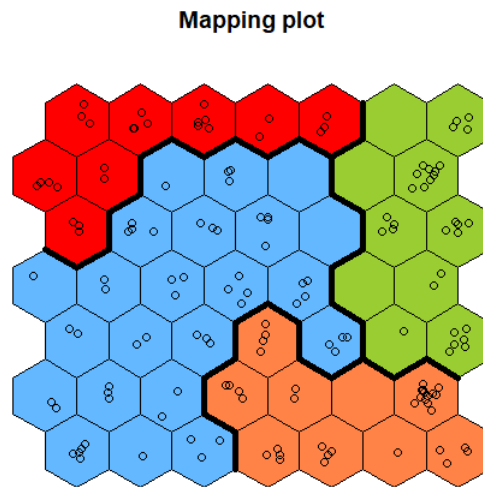


Figura 15: Cantidad de veces elegida una neurona como BMU y grupos hechos por la red SOM en una red 3x3

Si, por ejemplo, bajamos la cantidad de neuronas de la red a una red de 3x3, podemos ver como todas las neuronas han sido elegidas como BMU, y podemos apreciar los distintos grupos que genera la red.

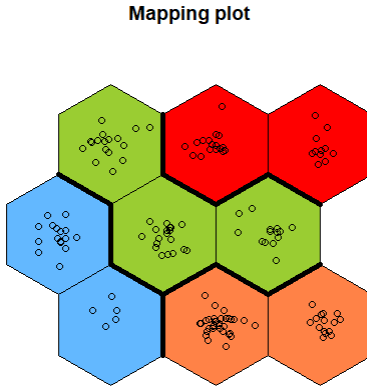


Figura 16: Cantidad de veces elegida una neurona como BMU y grupos hechos por la red SOM en una red 3x3

Con este ejemplo vemos la potencia de estas herramientas y el por qué de la elección de estas librerías, ya que nos permiten distintas representaciones de la red entrenada.

6. Exploración de *DataSets*

En general un *DataSet* es un conjunto de datos, en nuestro caso, se refiere a registros de voz, la forma más elemental estará compuesto por registros de voz (formato wav, mp4, etc), sin embargo, es muy común que los datos se almacenen con un preprocesado inicial después de aplicar algoritmos de extracción de características, en tal caso los datasets se encuentran en forma de tablas. Estos *DataSets* contienen conjunto de datos con características comunes. Estas características o *features* son representados como columnas del *DataSet*. Cada fila del *DataSet* es un dato completo, con tantas características como tenga el *DataSet*. En nuestro caso, una fila es un registro de voz, y una columna es una característica de ese registro de voz, como podría ser la frecuencia media, tonalidad... Normalmente estos datos se obtienen de una tabla, o la unión de varias tablas, de una base de datos, ya que siguen exactamente este mismo esquema, donde las columnas representan atributos comunes a las filas, y las filas son instancias de los datos.

Primeramente, debemos elegir el *DataSet* con el que entrenar el modelo y crear el mapa auto-organizado. Este es un paso crítico, ya que una buena selección del conjunto de datos puede suponer mejores resultados. Existen varias alternativas, estando sólo expuestas en las siguientes subsecciones aquellos *DataSets* que cumplen con la restricción de que posea datos de personas sanas y de personas con la EP. Esto es un requisito mínimo de los *DataSets* que se puedan considerar como candidatos, ya que no se estaría consiguiendo el objetivo si se entrenan los modelos con sólo datos de personas enfermas o sólo personas sanas, ya que no hay nada que diferenciar en grupos. El proceso que determinará el mejor candidato se basa en el cumplimiento de varios puntos clave. Estos puntos son:

- Fuente de los datos
- Indicador de nivel de desarrollo de la EP
- Disponibilidad de archivos WAV
- Cantidad de características
- Tipo de características
- Cantidad de registros
- Repetición del locutor
- Tamaño de grupos de locutores sanos y locutores enfermos

La calidad del *DataSet* vendrá determinado por cuántos puntos cumple o por la calidad del punto en cuestión. A mayor cantidad de puntos cumplidos, mayor calidad del *DataSet*. A continuación se pasará a hacer un resumen de los dos principales *DataSets* que encontramos para este proyecto.

6.1. Oxford Parkinson's Disease Detection *DataSet*

Este *DataSet*[30][31] fué desarrollado por la Universidad de Oxford en colaboración con el *National Centre for Voice and Speech*. Está compuesto por 195 registros, cada uno de ellos con 24 *features*. Entre estas *features*, se encuentran una mayoría respectivas a los registros de voz, pero otras como *Name* que hace referencia a un identificador del registro, y *Status* que significa si la persona está sana o padece de la EP, por lo que se puede utilizar dicho *DataSet* para algoritmos de aprendizaje supervisado o para aprendizaje no supervisado, como es en nuestro caso. En la mayoría de las características aparece las siglas MDVP, que significa Multi-Dimensional Voice Program, un programa de ordenador con el que se extrajeron algunas medidas de los registros de voz originales.

En cuanto a los registros, pertenecen a 31 personas diferentes, de las cuales 23 están diagnosticadas con la EP, y las otras 8 están sanas. Esto podría suponer un problema ya que los grupos parecen dispares y por lo tanto hay más registros de voz de personas con la EP.

```
import pandas as pd

df = pd.read_csv('D:\Databases\Telemonitoring\parkinsons.data.txt')
contSanos = 0
contEP = 0
for i in df['status']:
    if(i == 0):
        contSanos += 1
    else:
        contEP += 1
print("El numero de registros de personas sanas es de: " + str(contSanos) + " con un porcentaje de : "
      + str((contSanos/(contEP+contSanos))*100) + "% sobre el total (" + str(contEP+contSanos) + ")")
print("El numero de registros de personas con la EP es de: " + str(contEP) + " con un porcentaje de : "
      + str((contEP/(contEP+contSanos))*100) + "% sobre el total (" + str(contEP+contSanos) + ")")
```

Figura 17: Código para la obtención de la estadística

Ejecutando el código de la Figura 17, obtenemos el siguiente resultado expuesto en la Figura 18.

```
In [1]: runfile('D:/Databases/Telemonitoring/CodigoEstadistica.py', wdir='D:/Databases/Telemonitoring')
El numero de registros de personas sanas es de: 48 con un porcentaje de : 24.615384615384617% sobre el total (195)
El numero de registros de personas con la EP es de: 147 con un porcentaje de : 75.38461538461539% sobre el total (195)
```

Figura 18: Estadística del *DataSet*

Como podemos observar, la estadística confirma nuestras suposiciones: tenemos más registros de personas con la EP que sanas. Esta imagen revela que existen 48 registros de personas sanas frente a 147 de personas que padecen la EP. Esto podría suponer un problema, ya que implica que el *DataSet* está desbalanceado, pudiendo originar un mal entrenamiento ya que el modelo podrá generalizar más fácilmente la clase mayoritaria. Una posibilidad de igualar las clases sería aplicar un *Upsampling* a los datos de la clase minoritaria. El *Upsampling* es una técnica en la que se generan datos de manera sintética y son inyectados al *DataSet* para que quede una clase más pareja con el resto[35]. En el caso de este *DataSet*, no es recomendable al no tener la fuente original de los datos. Sin esta versión original de los datos, lo único que se conseguiría es añadir datos con ruido o con valores sin sentido y sin poder valorar si han sido generados de forma correcta. En cuanto a los puntos mencionados anteriormente, vamos a observarlos uno por uno para este *DataSet*.

■ Fuente de los datos

Este *DataSet* procede de la base de datos de la UCI antes mencionada, aunque originalmente fue

creada por la Universidad de Oxford en colaboración con el *National Centre for Voice and Speech*. Las fonaciones fueron extraídas en una sala insonorizada con un micrófono de diadema (AKG C420), a exactamente 8 centímetros de los labios. Este micrófono fue previamente calibrado con un fonómetro a 30 centímetros del orador. Las señales de la voz fueron directamente grabadas en un ordenador, muestreado a 44,1 kHz y con una resolución de 16 bits. Después se pasó a normalizar en amplitud los datos digitalmente, y entonces calcular las medidas. Al estar insonorizada la cabina, existe un menor ruido en las señales de voz, obteniendo registros o características con una mejor calidad. No obstante, puede que la normalización realizada distorsione los parámetros extraídos y afecte negativamente al detección de la EP.

- **Indicador de nivel de desarrollo de la EP**

En cuanto a este punto, no existe información al respecto, salvo el estado del paciente (0 para sano, 1 para persona con la EP). No es una mala métrica, pero podría estar acompañada de otras como el UPDRS.

- **Disponibilidad de archivos WAV** En este *DataSet* no se proporcionan los archivos WAV originales, teniendo sólo los datos ya procesados, por lo que no se podrán obtener otras características que no sean las ya contempladas en el *DataSet*.

- **Cantidad de características** En el siguiente Tabla 1 podemos observar las características del *DataSet*.

Característica	Descripción
Name	Nombre asociado al paciente y número de grabación
MDVP:Jitter(%)	Kay Pentax MDVP jitter expuesto como porcentaje
MDVP:Jitter(Abs)	Kay Pentax MDVP jitter en valor absoluto y en microsegundos
MDVP:RAP	Kay Pentax MDVP Amplitud relativa de la perturbación
MDVP:PPQ	Kay Pentax MDVP Cociente de perturbación periodica de 5 puntos
Jitter:DDP	Media en valor absoluto de las diferencias de las diferencias entre ciclos, dividido por el periodo promedio
MDVP:F0(Hz)	Promedio de la frecuencia vocal fundamental
MDVP:Fhi(Hz)	Máximo de la frecuencia vocal fundamental
MDVP:Flo(Hz)	Mínimo de la frecuencia vocal fundamental
MDVP:Shimmer	Kay Pentax MDVP Shimmer local, representa la variabilidad absoluta entre la amplitud de periodos consecutivos, divididos por la media total de la amplitud
MDVP:Shimmer(dB)	Kay Pentax MDVP Shimmer local en decibelios, representando la media en valor absoluto del logaritmo en base 10 de las diferencias entre amplitudes de periodos consecutivos, multiplicado por 20
Shimmer:APQ3	Cociente de la amplitud de la perturbación de 3 puntos
Shimmer:APQ5	Cociente de la amplitud de la perturbación de 5 puntos
MDVP:APQ	Kay Pentax MDVP Cociente de la amplitud de la perturbación de 11 puntos
Shimmer:DDA	Media en valor absoluto de las diferencias entre diferencias consecutivas entre las amplitudes de periodos consecutivos
NHR	Noise-to-Harmonics Ratio
HNR	Harmonics-to-Noise Ratio
RPDE	Recurrence Period Density Entropy
DFA	Detrended Fluctuation Analysis
D2	Dimensión de la correlación
PPE	Pitch Period Entropy
Status	0 para sano, 1 para persona con la EP
Spread1	Medida no lineal de la frecuencia fundamental de la variación
Spread2	Medida no lineal de la frecuencia fundamental de la variación

Tabla 1: Descripción de las características del Oxford Parkinson's Disease Detection *DataSet*

Aquí vemos que existen 24 características. No son una gran cantidad de características, por lo que a la hora de procesarlas, no tendremos que hacer un procesamiento muy extenso.

■ Tipo de características

Como observamos en el Tabla 1, podemos agrupar las características en 7 grupos. La distribución de grupos queda reflejada en la siguiente tabla:

Grupo	Características
Grupo de control	Status Name
Grupo de Variación de la Frecuencia Fundamental	MDVP:Jitter(%) MDVP:Jitter(Abs) MDVP:RAP MDVP:PPQ Jitter:DDP
Grupo de Variación de la Frecuencia Fundamental no lineales	Spread1 Spread2 PPE
Grupo de Variación de Amplitud	MDVP:Shimmer MDVP:Shimmer(dB) Shimmer:APQ3 Shimmer:APQ5 MDVP:APQ Shimmer:DDA
Grupo de Frecuencia Vocal Fundamental	MDVP:F0(Hz) MDVP:F1(Hz) MDVP:F2(Hz)
Grupo de Medidas de ruido en componentes tonales	NHR HNR
Grupo de Medidas de complejidad dinámicas	RPDE D2 DFA

Tabla 2: Descripción de grupos para el *Oxford DataSet*

La denotación *MDVP* hace referencia a *Multi-Dimensional Voice Program*, programa utilizado para extraer características de la voz.

■ Cantidad de registros

Como vimos en la Figura 18, nos encontramos ante un Dataset relativamente pequeño (195 entradas), aunque con suficientes registros como para poder entrenar una red de neuronas SOM.

■ Repetición del locutor

Los registros son únicos, pero varios de ellos pertenecen a una misma persona, pudiendo desenlazar en repetición de datos. Esto no tendría por que suponer un problema, ya que la red puede darle valor a atributos pertenecientes a personas con la EP independientemente de que pertenezca a la misma persona.

■ Tamaño de grupos de locutores sanos y locutores enfermos

En la estadística de la Figura 18 observamos que el *DataSet* está un poco desbalanceado. Al no poseer los archivos WAV, no podríamos realizar un *Upsampling* efectivo y sin añadir ruido a los datos. Esto puede parecer un problema, pero no. Este desbalanceo representa un entorno real, ya que solo un pequeño porcentaje de la población padece de la EP, lo que provoca que la red de neuronas SOM entrene de manera coherente.

Por todos estos puntos a favor, este *DataSet* podría ser una muy buena elección.

6.2. Multiple Sounds UCI *DataSet*

Este *DataSet* pertenece a la Universidad de California en Irvine (UCI), concretamente al repositorio de Machine Learning que poseen. En este repositorio podemos encontrar el *Parkinson Speech DataSet with Multiple Types of Sound Recordings Data Set*, acortado como Multiple sounds *DataSet*[39].

■ Training

Para el apartado de *training* se reunió a 40 personas, 20 de ellas sanas y 20 con la EP. A cada una de estas personas se les pidió que hablará especificándoles que decir. Los 3 primeros registros hacen referencia a 3 vocales distintas (a, o y u), los diez siguientes son los dígitos del 1 al 10, prosiguiendo con cuatro frases cortas y terminando con nueve palabras. Esto da un total de 1040 registros de voz para el entrenamiento de la red SOM (aunque en los datos aparecen 1039).

■ Test

El apartado de *test* es considerablemente más pequeño, lo que tiene sentido considerando que este *DataSet* solo se utilizará para verificar el modelo y no para entrenarlo. En este apartado sólo se contó con 28 personas que padecen la EP y sólo se les pidió articular dos vocales (a y o) tres veces, por lo que por cada persona tenemos solo 6 registros. En este caso, el total de registros es de 168 (aunque en los datos aparecerán 167).

```
import pandas as pd
import numpy as np

dfTraining = pd.read_csv('D:/Databases/MultipleSounds/train_data.txt')
dfTrainingnumpy = np.array(dfTraining)
print("-----Training-----")
print("Filas x Columnas",dfTrainingnumpy.shape)
contEPtraining = 0
contSanostraining = 0
for i in dfTrainingnumpy:
    if(i[28] == 0):
        contSanostraining += 1
    else:
        contEPtraining += 1
print("Num de personas con la EP: ",contEPtraining)
print("Num de personas sanas: ",contSanostraining)

print("-----Test-----")

dfTest = pd.read_csv('D:/Databases/MultipleSounds/test_data.txt')
dfTestnumpy = np.array(dfTest)
print("Filas x Columnas",dfTestnumpy.shape)
contEPtest = 0
contSanostest = 0
for i in dfTestnumpy:
    if(i[27] == 0):
        contSanostest += 1
    else:
        contEPtest += 1
print("Num de personas con la EP: ",contEPtest)
print("Num de personas sanas: ",contSanostest)
```

Figura 19: Código para la estadística del Multiple Sound *DataSet*

Ejecutando el código de la Figura 19 podemos observar la siguiente información.

```
-----Training-----
Filas x Columnas (1039, 29)
Num de personas con la EP: 519
Num de personas sanas: 520
-----Test-----
Filas x Columnas (167, 28)
Num de personas con la EP: 167
Num de personas sanas: 0
```

Figura 20: Información sobre los datos del *DataSet*

Como podemos ver en la Figura 20, el apartado de *training* está muy bien equilibrado, ya que hay la misma cantidad de registros de personas con la EP como personas sanas. En cambio el apartado *test* es el contrario, tenemos un *DataSet* totalmente desbalanceado al solo existir registros de personas con la EP. Analizando los datos, vemos que existe una diferencia en las columnas, y es que aparece una *feature* menos en el *DataSet* del test. Esto ocurre debido a que en el *DataSet* del test, no se tiene en cuenta el Unified Parkinson's disease Rating Scale (UPDRS), siendo una medida para cuantificar el avance de la EP. Esta diferencia puede ser obviada ya que no debería entrenarse el modelo con esta *feature* al tratarse de un valor asociado directamente con la EP, y no con características de la voz. El resto de *features* son comunes a ambos *DataSet*, estando entre ellas las distintas versiones del *Jitter* (local, absoluto, rap, ppq5 y ddp) y del *Shimmer* (local, dB, apq3, apq5, apq11 y dda), antes mencionados en el apartado *Caracterización de la voz*. En la siguiente tabla se representan todas las características.

Característica	Descripción
SubjectId	Identificador del registro
Jitter(local)	Variabilidad absoluta de la frecuencia periodo a periodo, dividido por la frecuencia media total de los periodos de la señal
Jitter(local,absolute)	Diferencia absoluta media entre periodos consecutivos
Jitter(RAP)	Variabilidad relativa de la frecuencia entre un periodo y la media de los dos periodos contiguos, dividido por la frecuencia media total de los periodos de la señal
Jitter(ppq5)	Cociente de perturbación periodica de 5 puntos
Jitter(ddp)	Media en valor absoluto de las diferencias de las diferencias entre ciclos, dividido por el periodo promedio
Shimmer(local)	Variabilidad absoluta entre la amplitud de periodos consecutivos, divididos por la media total de la amplitud
Shimmer(local,db)	Media en valor absoluto del logaritmo en base 10 de las diferencias entre amplitudes de periodos consecutivos, multiplicado por 20
Shimmer(apq3)	Cociente de la amplitud de la perturbación de 3 puntos
Shimmer(apq5)	Cociente de la amplitud de la perturbación de 5 puntos
Shimmer(apq11)	Cociente de la amplitud de la perturbación de 11 puntos
Shimmer(dda)	Media en valor absoluto de las diferencias entre diferencias consecutivas entre las amplitudes de periodos consecutivos
AC	Autocorrelation
NTH	Noise-to-Harmonics
HTN	Harmonics-to-Noise
Median pitch	La mediana del tono
Mean pitch	Tono medio
Standard deviation	Desviación estandar del tono
Minimum pitch	Tono más bajo
Maximum pitch	Tono más alto
Number of pulses	Número de pulsos del tono del paciente
Number of periods	Número de periodos en el registro de voz
Mean period	Media del periodo
Standard deviation of period	Desviación estandar del periodo
Fraction of locally unvoiced frames	<i>Frames</i> de la voz en los que no se registra la voz debido a que el paciente no puede mantener la voz durante la fonación
Number of voice breaks	Número de veces que la voz se corta durante un periodo de tiempo
Degree of voice breaks	Duración total del corte en la voz entre la duración total de la grabación
UPDRS	Medida del nivel de la EP del paciente
Class information	Determina si el paciente esta sano o padece de la EP

Tabla 3: Características del *DataSet* totales

Ahora procedemos a analizar los puntos clave por los que filtraremos los *DataSets*.

- **Fuente de los datos**

Este DataSet pertenece a un estudio del Departamento de Neurología en *Cerrahpas*, una facultad de medicina de la *Istanbul University*. Este estudio se realizó sobre 20 personas enfermas con la EP (6 mujeres, 14 hombres) y 20 personas sanas (10 mujeres, 10 hombres) para la parte de *Training*. En cuanto a la parte de *Training*, consistió de un grupo de pacientes con la EP que tenían la enfermedad diagnosticada desde hace 6 años como máximo, mientras que para el *Test* eran pacientes con la EP diagnosticada como máximo hace 13 años, a fecha de realización del estudio. Las grabaciones se obtuvieron a través de un micrófono Trust MC-1500 con un rango de frecuencias de entre 50Hz y 13kHz. El micrófono estaba configurado a 96kHz, 30dB y a una distancia de 10 centímetros del paciente. Tanto para el *Training* como para el *Tests*, los registros se obtuvieron con esta configuración. Este es un buen punto a favor de este *DataSet* ya que la configuración es la misma para todos los pacientes, teniendo todo los registros aproximadamente el mismo ruido, aunque no el mismo ya que no hay datos de que se realizase en unas condiciones óptimas de insonorización.

- **Indicador de nivel de desarrollo de la EP**

Como hemos visto, la característica 28 hace referencia al UPDRS, que como hemos dicho anteriormente, es una medida del nivel de la EP del paciente. Esto es un gran punto a favor ya que permite a la red SOM crear el mapa distribuyendo mejor los datos por dicha característica, lo que permite generar mejores grupos en el mapa. De igual modo, se pueden generar grupos sin esta característica.

- **Disponibilidad de archivos WAV** En este *DataSet* disponemos de los archivos .WAV de los audios, aunque la adquisición de nuevas características y el procesado de los archivos .WAV no están incluidos en los objetivos de este trabajo.

- **Cantidad de características**

Como habíamos dicho con anterioridad, el *DataSet* se divide en *Training* y *Test*, teniendo las mismas características salvo el UPDRS para el *Test*. El total de características es de 29 para el *Training* y 28 para el *Test*.

- **Tipo de características** El *DataSet* se puede dividir en 2 grandes grupos y un tercero más minoritario. El primero de ellos estaría formado por todas las características relacionadas con la frecuencia y la amplitud (características 2-12). El segundo estaría relacionado con el tono y el periodo (características 16-24). Por último, las características 25, 26 y 27 que están relacionadas con fallos en la voz durante la grabación. El resto de características se podrían denominar características de control de los registros y de los pacientes (UPDRS, SubjectId y Class Information. Estas características son bastante parecidas a las del *DataSet* anterior, siendo estas también objetivas y relacionadas con el objetivo del proyecto. Al igual que en el *DataSet* anterior, vamos a organizarlas en grupos en una tabla.

Grupo	Características
Parámetros de frecuencia	Jitter(local) Jitter(local,absolute) Jitter(RAP) Jitter(PPQ5) Jitter(DDP)
Parámetros relacionados con los pulsos	Number of pulses Number of periods Mean period Standard dev. of period
Parámetros de amplitud	Shimmer(local) Shimmer(local,dB) Shimmer(APQ3) Shimmer(APQ5) Shimmer(APQ11) Shimmer(DDA)
Parámetros de defectos en el registro de voz	Fraction of locally unvoiced frames Number of voice breaks Degree of voice breaks
Parámetros tonales	Median pitch Mean pitch Standard deviation Minimum pitch Maximum pitch
Parámetros armónicos	AC NTH HTN

Tabla 4: Descripción de grupos para el *Multiple Sounds DataSet*[39]

■ Cantidad de registros

En cuanto al número de registros, encontramos 1039 registros para el *Training* y 167 para el *Test*, manteniendo una relación un poco baja entre los datos del *Test*(13,85 %) respecto a los del *Training*(86,15 %). Esto puede solventarse transfiriendo algunos datos del apartado de *Training* al de *Test* para alcanzar el 25 %.

■ Repetición del locutor

Como hemos visto, los participantes del estudio del que se obtuvo este *DataSet*, tuvieron que hablar repetidas veces, teniendo cada uno de ellos 26 registros para el *Training* y 6 para el *Test*, teniendo bastantes datos repetidos, ya que el tono medio, por ejemplo sería siempre el mismo para los registros de una misma persona.

- **Tamaño de grupos de locutores sanos y locutores enfermos** En la imagen anterior (Figura 20) podemos ver como el apartado de *Training* está perfectamente balanceado, mientras que en el *Test*, sólo existen registros de personas con la EP, lo que puede suponer un problema ya que tendríamos que transferir datos del *Training* al *Test*, desbalanceándolo.

Este *DataSet* es una peor opción que el anterior.

6.3. Elección de *DataSet*

Como acabamos de ver, las dos opciones de *DataSet* tienen puntos a favor y puntos en contra, aunque la decisión parece clara, el Oxford Parkinson's Disease Detection *DataSet* es la mejor elección. Este será el *DataSet* con el que se crearán todos los modelos de redes SOM.

7. Procesamiento y modelado

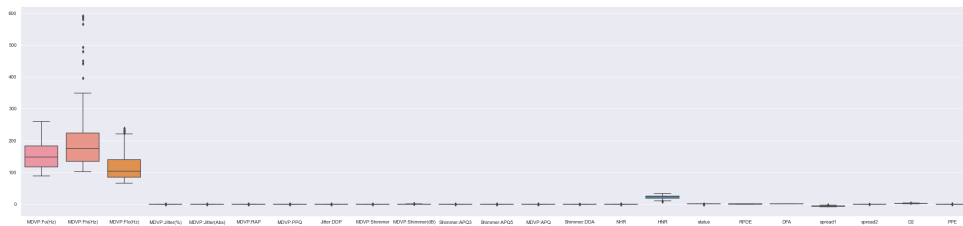
7.1. Introducción

Después de explicar el contexto del trabajo y elegir el *Oxford Parkinson's Disease Detection DataSet* [30][31], pasaremos a analizar los registros, comprobar que no hay datos nulos, ver datos atípicos o *Outliers*, sacar gráficas de cada características... Este análisis profundo se hará en pro a mejorar y limpiar el *DataSet* con los que entrenaremos los modelos. Estos modelos serán únicos, ya que en cada uno de ellos se cambiarán el número de neuronas de las que consta o el grupo de características en el modelo.

7.2. Exploración y análisis de los datos

Al explorar los datos, lo primero que tenemos que comprobar es la existencia de datos nulos, ya que estos valores suponen un problema en el caso de existir. El resultado es de un total de 0 valores nulos en todo el *DataSet*. Una vez comprobado esto, pasamos a ver la distribución de las densidades de las características y el diagrama de bigotes o *Boxplot*, con lo que conseguiremos observar que distribución siguen y si se tienen que normalizar los datos por estar en escalas muy distantes. La mayoría de las características siguen distribuciones similares a una Distribución Normal, pero el rango de valores si que es distinto, algunos siendo valores muy pequeños (del orden de centésimas), otras características con sólo valores negativos y otros muy grandes comparado con el resto (con valores incluso mayores a 250). Para ver que realmente siguen una Distribución Normal, vamos a utilizar una herramienta que nos indicará cuánta verdad hay en dicha suposición obtenida a través de la exploración de los datos. Esta herramienta es el *Kurtosis Test*, que está basado en la medida de *Kurtosis*, que indica como de parecido se halla un conjunto de datos a una distribución normal [36]. Esta herramienta se encuentra en la librería de *scipy*, no mencionada anteriormente ya que sólo se utiliza esta herramienta en todo el desarrollo. El resultado se encuentra en el fichero *Kurtosis.txt*, y se observan que los valores se encuentran muy cercanos al 0, indicando que su parecido a una distribución Normal es muy alta. Los valores negativos indican que la gráfica de densidad de la característica es más achatada, y cuanto mayor es el valor en positivo, mayor grado de inclinación tiene la pendiente de la gráfica.

En cuánto a los valores atípicos, podemos ver por cada columna las estadísticas relativas a estos datos (rango intercuartil, límite superior, límite inferior, porcentaje de *Outliers* y cantidad total de *Outliers*) gracias a la función *outliers*. Estos valores también están representados de manera gráfica en los diagramas de bigotes, pero para analizarlos mejor, necesitamos de esta estadística. Estos parámetros se guardan en un fichero llamado *Outliers.txt*, generado por el propio código. En el vemos que no existe una cantidad excesivas de *Outliers*, siendo la característica *NHR* donde se encuentran más *Outliers* con un total de 19. Es importante tener localizados estos datos al poder influir de manera negativa al entrenamiento de la red de neuronas auto-organizada.



del real a los datos con números más altos. Para que todos los datos estén en la misma escala debemos aplicar una herramienta de estandarización. Existen varias en Python, como mencionamos antes en el apartado de Librerías. En el caso de la siguiente figura, se aplicó un *StandardScaler*, que distribuye los datos de cada columna siguiendo una Distribución Normal(0,1).

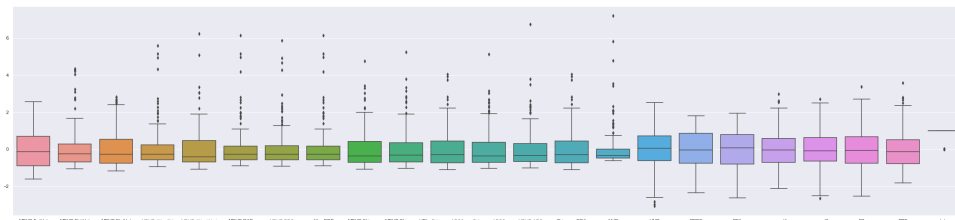


Figura 22: Representación de todas las características en diagramas de bigotes una vez aplicado el *StandardScaler*

Como vemos en la Figura 22, ahora todas las características se encuentran en el mismo rango de valores por lo que se puede apreciar mejor los *Outliers* y como características como la de NHR poseen una cantidad más alta de valores en los *Outliers*. Esta imagen se incluye en el Anexo 1 para su mejor visualización. Una vez aplicado el *StandardScaler* a las características, podemos dibujar en la misma gráfica todas las distribuciones de densidad solapadas, observando la similitud de distribuciones entre estas características.

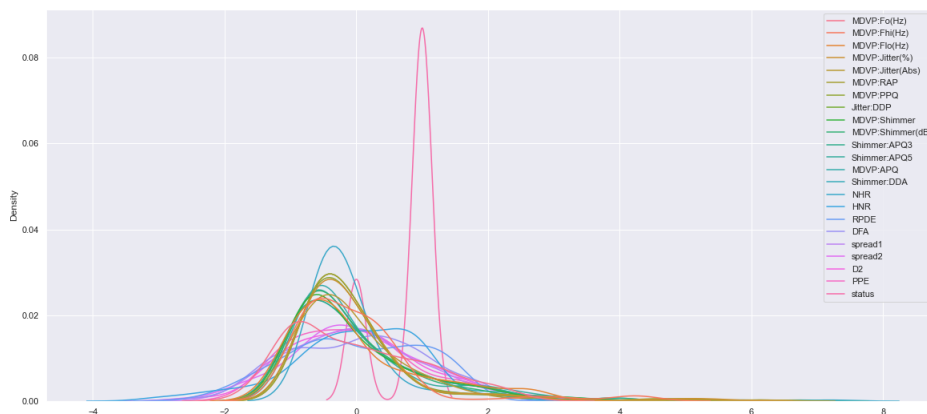


Figura 23: Representación de todas las características en diagramas de densidad solapados una vez aplicado el *StandardScaler*

Cabe destacar que existe una característica que es más distinta que el resto, siendo esta *status*. Esto es normal ya que dicha característica solo toma 2 valores, 0 o 1. En cambio, las demás características parecen seguir la misma tendencia, entre el 0 y el -1 se encuentran la mayoría de sus valores, y progresivamente se van reduciendo según se avanza en cualquiera de los dos sentidos del eje X. Esta imagen se incluye en el Anexo 1 para su mejor visualización. Teniendo en cuenta esta gráfica y el archivo *Kurtosis.txt* presentes, se puede observar como las características se asemejan bastante a una distribución Normal. Por ejemplo la característica de color azul claro, y con la cima de la distribución más alta (sin contar con *status*), es NHR, teniendo un parecido muy alto a una distribución normal, y si nos vamos al archivo *Kurtosis.txt* observamos como este valor es tan cercano a 0 que se podría considerar igual a 0. En la siguiente tabla viene representada la información guardada en dicho archivo.

Característica	Valor del KurtosisTest
MDVP:Fo(Hz)	0.011875117260038788
MDVP:Fhi(Hz)	4.935096395062021e-10
MDVP:Flo(Hz)	0.09330047673931735
MDVP:Jitter(%)	1.0442039617840122e-12
MDVP:Jitter(Abs)	4.242378718086615e-12
MDVP:RAP	1.023807024596679e-13
MDVP:PPQ	1.1282402248250713e-12
Jitter:DDP	1.012827962934764e-13
MDVP:Shimmer	9.571535278396596e-06
MDVP:Shimmer(dB)	6.875091449382297e-08
Shimmer:APQ3	4.767756297867252e-05
Shimmer:APQ5	1.566943995797579e-06
MDVP:APQ	2.9376839379462037e-12
Shimmer:DDA	4.75993753388172e-05
NHR	2.273715938291115e-16
HNR	0.10820118650767349
status	0.02029177767388102
RPDE	1.0181747174420626e-06
DFA	0.003826311804631507
spread1	0.9975246934587328
spread2	0.9145519102974735
D2	0.4529561866931672
PPE	0.150960893140275

Tabla 5: Valores del KurtosisTest por característica

Como dijimos anteriormente, se crearán varios modelos de redes SOM con distintos grupos de características, pero para saber qué grupos hacer, necesitamos saber que características están relacionadas entre ellas, para poder elegir características que no estén muy relacionadas entre ellas, ya que si no, tendríamos varias columnas representando la misma información, implicando que el aprendizaje de la neurona vaya peor de lo esperado. Para ello, vamos a representar una matriz de correlación, usada para determinar el grado de relación entre las características del *DataSet*. Cabe destacar que dicha matriz de correlación se obtiene a través del coeficiente de correlación de *Pearson*, que implica la suposición de linealidad sobre las características[29].

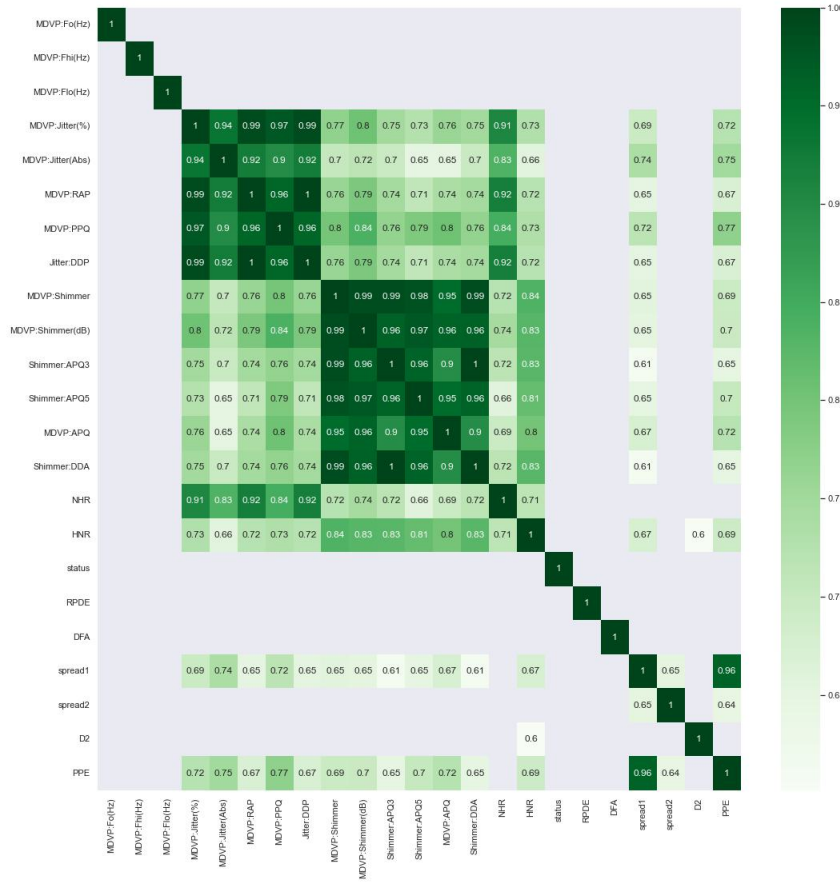


Figura 25: Matriz de correlación del *DataSet* filtrada por valores con correlacion ≥ 0.6

Como podemos observar en esta Figura 25, no se encuentra ninguna correlación alta entre la característica *Status* y el resto, teniendo que escoger grupos de características para entrenar los modelos de forma obligatoria, al no haber relación directa entre las características. A parte de la matriz de correlación, existen otros métodos para extraer grupos de características con los que entrenar los modelos. Uno de ellos es el *Feature Selection*. Esta funcionalidad permite elegir las mejores características respecto a una diana, en nuestro caso la característica *Status*. En la siguiente gráfica de la Figura 26, veremos como se han valorado las características a través del *Feature Selection*, ordenadas de menor a mayor puntuación. En el eje Y se representa dicho valor, y en el eje X la característica con dicho valor.

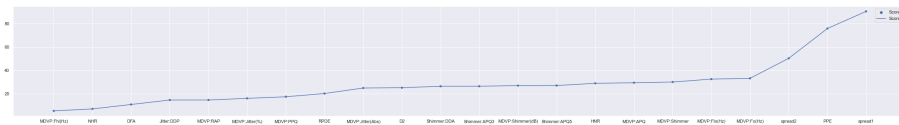


Figura 26: Puntuación de las características según el *Feature Selection*

Esta Figura 26 nos permite, al igual que la matriz de correlación, hacer grupos de características y seleccionar de cada grupo las características de mayor valor. Pero, no es la única herramienta que puntúan

las características. La *Mutual Information* permite puntuar a las características a través de su relación con la característica objetivo. Esta herramienta mide la dependencia mutua de las dos variables, a través de la reducción de la incertidumbre de las variables debido al conocimiento de otra variable, en nuestro caso la variable objetivo *status*. La *Mutual Information* es una herramienta parecida, pero no igual, a la matriz de correlación, pero exclusivamente con una variable objetivo.

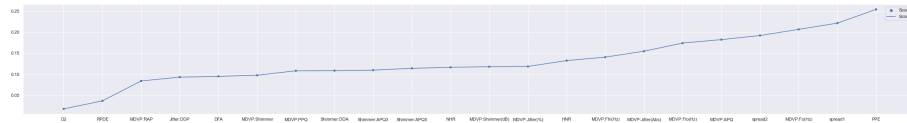


Figura 27: Puntuación de la *Mutual Information* del *DataSet*

Si comparamos esta Figura 27 y la Figura 24, podemos ver que para la columna *status* los valores no se corresponden, ni en valor, ni en posición de la ordecaión de menor a mayor. Esto se puede ver en las características, por ejemplo, *PPE* y *spread1* donde se encuentran en primer lugar y segundo lugar respectivamente en la Figura 27, mientras que en la Matriz de correlación (Figura 24) se encuentran en segundo lugar y primer lugar, respectivamente. Tampoco se corresponde con la Figura 26, correspondiente a la gráfica de la herramienta *Feature Selection*, por lo que tendremos que tener en cuenta a la hora de formar grupos si estamos utilizando esta información sobre la *Mutual Information*.

Una vez recopilada toda esta información podemos pasar a agrupar las características en grupos con los que se entrenarán las redes SOM. Para ello, primeramente debemos asociar un identificador a cada grupo, creando un código para los identificadores con una expresión regular con cierta carga semántica, para tener información sobre el origen del grupo, por lo que se tratará de una clave semántica. La expresión regular será la siguiente, y se pasará a explicar que significa cada letra de la expresión regular:

$$G[0-9]^+ _ NE _ [0-9]^+ _ [0-9]^+ x [0-9]^+$$

donde un símbolo elevado a un '+' significa la aparición una o más veces de ese símbolo. Como vemos, la expresión regular esta compuesta de 4 grupos, separados entre ellos con '_'. Como cada uno de ellos contiene cierta carga semántica, vamos a pasar a explicar que significa cada apartado.

- **G[0-9]** representa el número de grupo. Este apartado se ha puesto para diferenciar rápidamente a los grupos cuando se esté llevando a cabo un entrenamiento de una red SOM y se quiera hacer referencia a este grupo.
- **NE** representan dos atributos distintos, con distintos valores cada uno. Por una parte, N representa el tipo de método de estandarización, teniendo diferentes valores dependiendo del método usado. Los valores se representan en la siguiente tabla:

Valor de N	Método de estandarización
M	Método de <i>Sklearn MinMaxScaler</i>
R	Método de <i>Sklearn RobustScaler</i>
S	Método de <i>Sklearn StandardScaler</i>
A	Método de <i>Sklearn MaxAbsScaler</i>

Tabla 6: Especificación del valor de N según el método de estandarización elegido

Por otra parte, E representa el método de elección de características que se ha usado para formar dicho grupo de características, teniendo dos opciones comentadas anteriormente, representadas en la siguiente tabla:

Valor de E	Método de Elección
F	<i>Feature Selection</i>
C	Matriz de correlación
I	<i>Mutual Information</i>
FC	Combinación de ambos métodos <i>Feature Selection</i> y matriz de correlación
IC	Combinación de ambos métodos <i>Mutual Information</i> y matriz de correlación

Tabla 7: Especificación del valor de E según el método de elección de características

La combinación de ambos métodos puede aplicarse primero eligiendo los grupos por *Feature Selection* y después filtrar gracias a las correlaciones de la matriz de correlación, o al revés. Lo mismo ocurre con *Mutual Information* y la matriz de correlación.

- [0-9] hace referencia a la cantidad de características que componen el grupo. Este número puede repetirse en varios identificadores, aunque hagan referencia a distintas características.
- [0-9]x[0-9] es el tamaño de la red que entrenaremos con este grupo. Cada número representa la cantidad de neuronas que tiene dicho eje. La x se utiliza para diferenciar que número pertenece a cada eje.

7.3. Preprocesamiento

Una vez habiendo explorado los datos y habiendo creado una notación para los identificadores de los grupos de características faltan crear los grupos con los que se entrenarán los distintos modelos de redes de neuronas SOM. Para ello, primeramente debemos tener en cuenta que la Matriz de correlación de la Figura 25 nos aporta información sobre qué características están relacionadas, por lo que debemos analizar la matriz para obtener información sobre qué características no deberían ir en el mismo grupo, ya que sería duplicar información para la red.

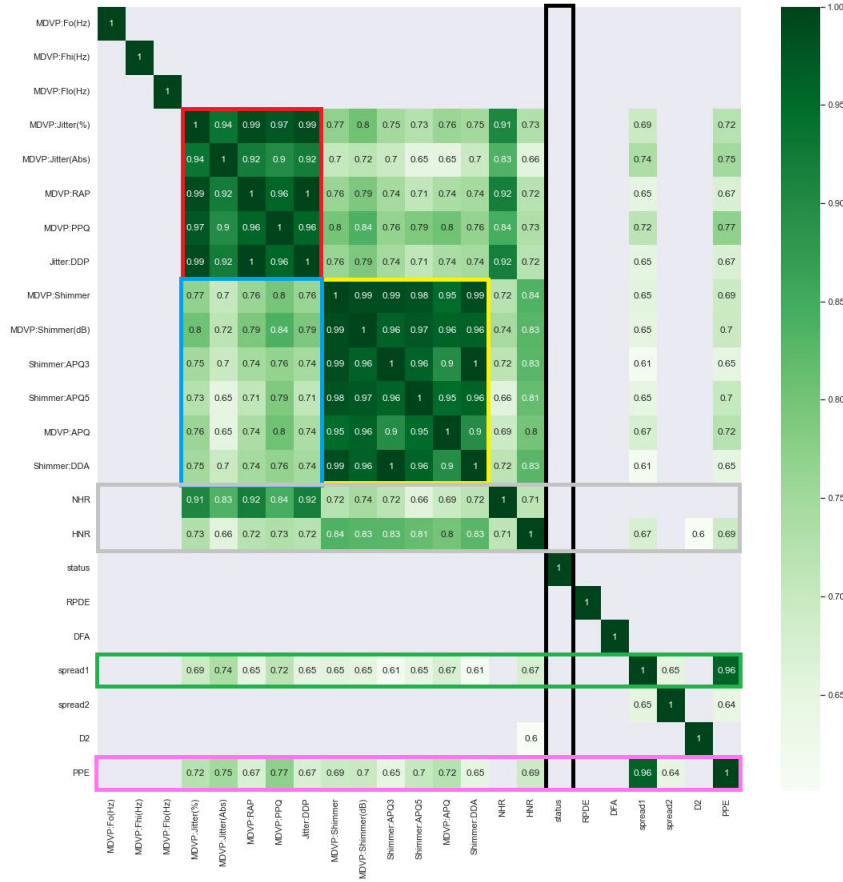


Figura 28: Matriz de correlación filtrada, por correlación mayor o igual a 0.6, con grupos de correlación

En esta Figura 28, vemos como estan bordeadas las características que se refieren a ellas mismas, siendo estas los dos grandes grupos con colores amarillo y rojo, representando las características de *Shimmer* y *Jitter*, respectivamente. A estos grupos de características se les pasará a llamar grupo de correlación, para que no exista duda entre grupo de correlación cuando nos refiramos a los grupos a partir de esta imagen con los bordes de colores, y a los grupos de características, referentes a los grupos de las características con los que entrenarán los modelos. Como se puede apreciar en estos grupos con borde rojo y amarillo, todos los valores de correlación superan el 0.9, siendo valores muy altos de correlación. Con estos datos se puede determinar que en los grupos de características no deberían existir varias características referentes a estos grupos de correlación, aunque debería probarse mínimo un grupo de características con varias pertenecientes al mismo grupo de correlación, ya sea amarillo o rojo, para corroborar esto. Esto debe hacerse ya que correlación no siempre implica causalidad[29]. Esto significa que aunque parezca que X implica Y, puede ser que realmente Z implique X e Y, incluso que sea cuestión de azar dicha correlación[29]. Aunque los valores de correlación sean altos, no implica la no inclusión imperativa de dichas características por lo anterior dicho, ya que también debemos valorar la naturaleza de las características, como en este caso, al pertenecer a la misma naturaleza de la característica, no debería repetirse información, es decir, incluir varias características referentes provenientes de la misma naturaleza. Por otro lado vemos un grupo de correlación, bordeado con color azul, donde se establece una correlación alta entre *Shimmer* y *Jitter*. Justo debajo de los bordes amarillo y azul vemos el borde gris, rodeando las características HNR y NHR. Este grupo de correlación muestra que,

dichas características referentes al ruido, tienen bastante correlación con el resto, especialmente con el *Jitter*. Esta correlación es negativa, es decir, cuánto más alto es el valor del ruido, menor es el valor del *Jitter* y *Shimmer*. El ruido parece una característica esencial ya que podría tener mucha relación con la EP al estar tan relacionada con las demás características. También se han bordeado con lila y verde las filas referentes a *spread1* y *PPE*. Se ha destacado en la imagen con color negro los bordes de la característica *status* para observar que ninguna característica guarda una relación alta con ella, restringiendo la creación de grupos de una sola característica, y obligando a que sean un conjunto de características las que formen los grupos de características. Las características con toda la fila en gris quiere decir que no guarda una correlación alta con el resto de características y no forma un grupo de correlación, por lo que deberían estar en todo grupo de características. Elegir una característica, o varias, de estos grupos de correlación significa crear tantos grupos como combinaciones posibles existan entre las elecciones de características de los grupos. Esto genera enormes cantidades de grupos de características, inabarcable en un proyecto como este, por lo que se elegirán arbitrariamente grupos con dichas características, cumpliendo las suposiciones anteriormente planteadas.

7.4. Construcción de los modelos

Ahora nos disponemos a crear los modelos con los identificadores de los grupos anteriormente explicados. Estos modelos se llamarán experimentos, debido a su naturaleza, al no saber si el resultado de cada uno de ellos será un resultado positivo o negativo, aunque independientemente del resultado se podrán obtener conclusiones sobre dichos experimentos. Los primeros harán referencia al mismo grupo de características. Esto se ha planteado así para primeramente realizar diversos cambios sobre los parámetros disponibles por la librería que manipula la red de neuronas. Si cambiasemos el grupo de características y a la vez los valores de los parámetros, no podríamos asegurar si los resultados son debido a los valores de los parámetros o a las características elegidas. Una vez realizado dichos experimentos sobre los parámetros, se elegirán los que parecen obtener una mejor dinámica.

7.4.1. Experimento 1: G01_SC_22_10x10

Primeramente pasaremos a explorar el primer grupo de todos, con todas las características. Este grupo, denominado **G01_SC_22_10x10**, será generado a partir de Python, como archivo .CSV para después servir como punto de partida para el entrenamiento de la red SOM en R. Se ha elegido empezar por este tamaño de red (10x10) debido a la incertidumbre del primer entrenamiento de una red SOM, ya que es un tamaño ni excesivamente pequeño como para que la red no pueda generar varios grupos, ni suficientemente grande como para que el ratio registro/neurona sea menor a 1, incurriendo en posibles errores de entrenamiento con una cantidad alta de neuronas sin ser elegidas como BMU. Todos los parámetros que se van a usar en la red de neuronas se mantendrán por defecto, para posteriormente, poder realizar comparativas con los resultados de los experimentos de las redes con parámetros cambiados. Estos parámetros son los siguientes:

- **Rlen**

Este parámetro indica la cantidad de *Epochs* que se realizarán en la red de neuronas. Por defecto este valor es de 100 pero, al igual que en el ejemplo, usaremos de base un valor de 1000.

- **Alpha**

Significa el ratio de aprendizaje de la red. Este valor por defecto empieza en 0.05, y llega a su mínimo valor cuando la red realiza su último *Epoch*, siendo este valor 0.01.

- **Radius**

Radius hace referencia al radio de vecindad de cada neurona. Este radio por defecto empieza abarcando un tercio de las neuronas de la red, y acaba por sólo afectar a la neurona elegida como BMU.

El archivo *G01.CSV* contendrá los datos ya estandarizados, en este caso a través del *Standard Scaler*. Una vez hecho esto, podemos empezar el entrenamiento de la red SOM. Al ser la primera vez que entrenamos

una red SOM, tendremos que elegir qué tipo de inicialización se hará antes de empezar con el entrenamiento. Hay estudios que recomiendan la elección de la inicialización aleatoria[3], pero si elegimos ésta, debemos asegurarnos de que los resultados no dependen de dicho azar. Para ello, repetiremos el proceso 4 veces, observando si existe algún cambio en los resultados, o no. Justo en esta parte, entra la librería anteriormente mencionada *aweSOM*, permitiéndonos lograrlo.

El proceso que realizaremos es muy parecido al del ejemplo descrito en el apartado del lenguaje R, pero para conseguir la aleatoriedad deseada para comprobar si convergen los distintos intentos, tenemos que comentar el comando que impone una semilla de aleatoriedad. Una vez probado varias veces, se concluye que no convergen exactamente a los mismos resultados, pero si patrones en todos ellos. En los 4 intentos se puede comprobar que en una de las esquinas se encuentran neuronas más aisladas con respecto al resto, pudiendo observarse también como todas generan prácticamente los mismos *clusters* a la hora de crear grupos. Esta característica tiene una gran influencia en la creación de los *clusters*, viendo como tanto la creación de 2, 3 y 4 *clusters* comienzan en la esquina donde las neuronas se encuentran a una mayor distancia entre ellas. En la siguiente Figura 29 se puede observar las distancias de las neuronas sobre las distintas redes entrenadas.

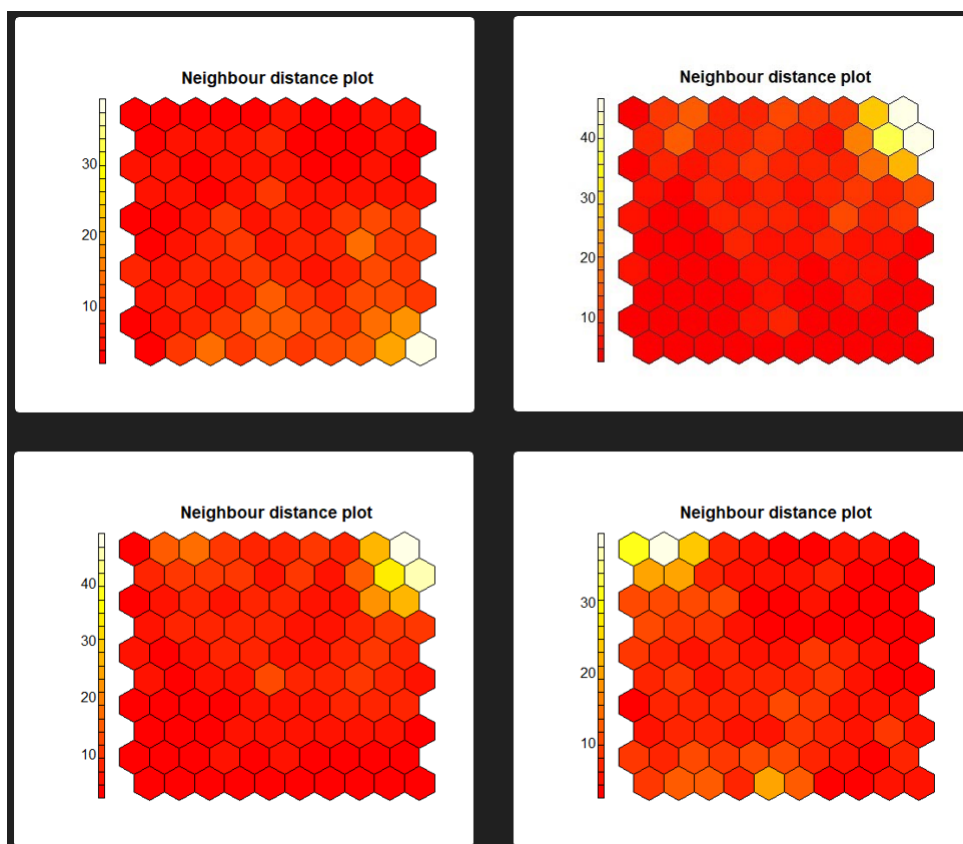


Figura 29: Distancias entre las neuronas de las 4 redes formadas con distintas *seeds*

Aun observando estos patrones, el comando `set.seed(23)` es necesario para que sean reproducibles los resultados. Esta línea establece un punto de partida para los números aleatorios generados, por lo que siempre se generarán los mismos números. Una vez establecida la semilla, con el código del Anexo X, podemos extraer las siguientes gráficas:

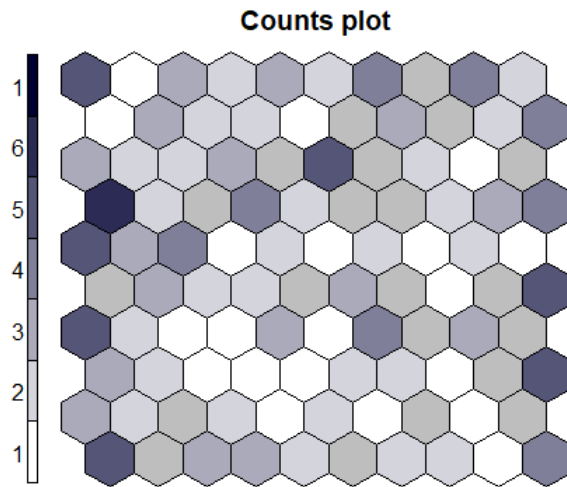


Figura 30: Número de elecciones como BMU de cada neurona de la red

Podemos observar en la Figura 30 que lo mayores números de BMU se ubican en los laterales de la red, dejando el interior de la red más vacío¹.

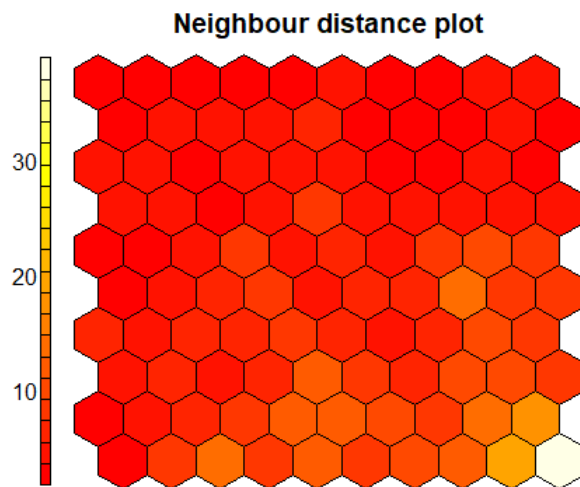


Figura 31: Distancias entre las neuronas de la red con sus neuronas vecinas

Vemos que prácticamente todas las neuronas se encuentran a una distancia cercana a sus vecinas, salvo en la esquina inferior derecha, donde se encuentran muy distantes. En las imágenes posteriores veremos como

¹Cabe destacar que en la leyenda se solapa un número, siendo un problema de la librería de R, suponiendo que en vez de aparecer 10 en la parte más alta de la leyenda, aparezca un 1

el algoritmo interpreta que es en esa zona donde no existe tanta relación con el resto de neuronas, por lo que empezará a crear *clusters* por dicha zona.

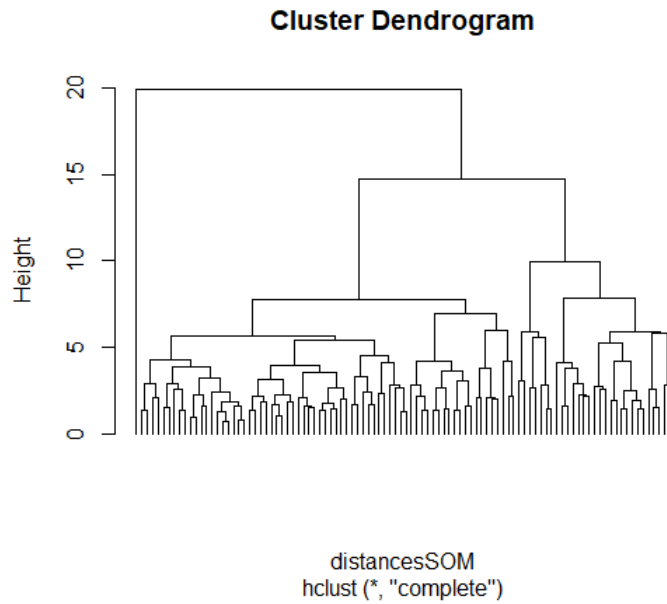


Figura 32: Árbol de relaciones jerárquicas o dendrograma de los *clusters*

En esta vemos como el algoritmo junta neuronas para crear grupos. Las neuronas vienen representadas en el eje X, y a medida que avanzamos en el eje Y se van juntando en distintos *clusters*. Según el número de *clusters* que le pidamos crear al algoritmo, realizará un corte u otro. En las Figuras 33, 34 y 35 podemos observar los distintos cortes que realizaremos, indicando que queremos 2, 3 y 4 *clusters*, respectivamente. Cada uno de ellos está representado por diferentes colores, y esto se hace para poder analizar si el algoritmo forma grupos con un mayor porcentaje de acierto sobre la variable *status*.

Mapping plot

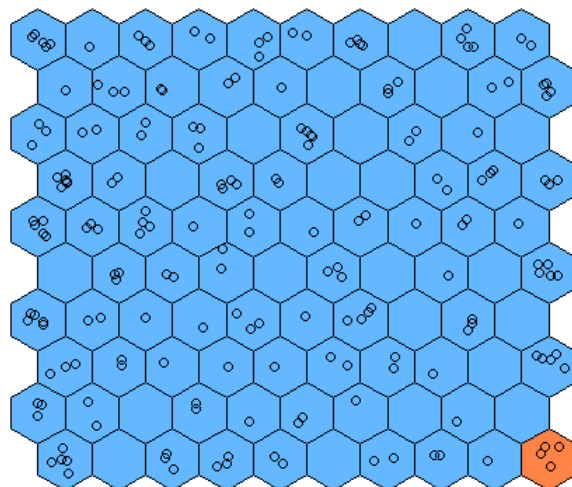


Figura 33: Grupos formados por la red para la imposición de 2 grupos

Mapping plot

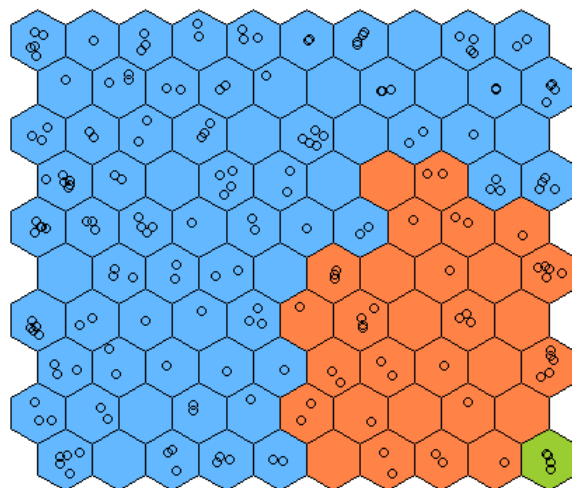


Figura 34: Grupos formados por la red para la imposición de 3 grupos

Mapping plot

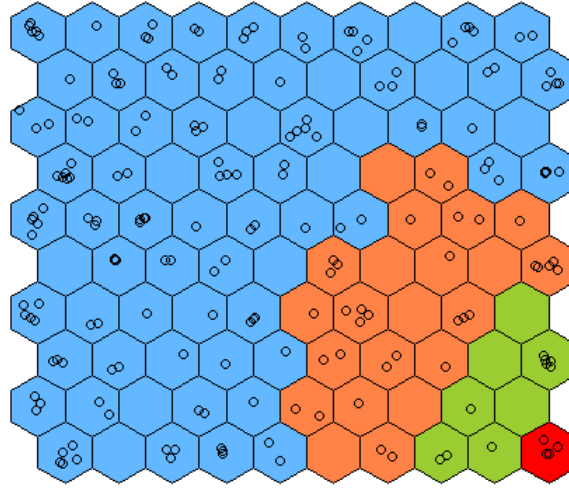


Figura 35: Grupos formados por la red para la imposición de 4 grupos

Una vez agrupadas las neuronas, debemos ver si estos grupos tienen alguna relación con la variable *status*, que determinaba si la persona padecía la EP o no. Para ello, utilizaremos la representación numérica de estos grupos, siendo el mayoritario el grupo 1, seguido del grupo 2, 3 y 4, respectivamente a la cantidad de neuronas que forman los grupos. Esta información se almacena en el archivo *G01.txt*, pero representaremos parte de su información en la siguiente Tabla 8.

Nombre del registro	Neurona	status	NClusters = 2	NClusters = 3	NClusters = 4
phon_R01.S01_1	60	1	1	2	2
phon_R01.S01_2	50	1	1	2	2
phon_R01.S01_3	50	1	1	2	2
phon_R01.S01_4	50	1	1	2	2
phon_R01.S01_5	50	1	1	2	2
phon_R01.S01_6	50	1	1	2	2
phon_R01.S02_1	91	1	1	1	1
phon_R01.S02_2	91	1	1	1	1
phon_R01.S02_3	99	1	1	1	1
phon_R01.S02_4	99	1	1	1	1
phon_R01.S02_5	99	1	1	1	1
phon_R01.S02_6	99	1	1	1	1
phon_R01.S04_1	94	1	1	1	1
phon_R01.S04_2	74	1	1	1	1
phon_R01.S04_3	84	1	1	1	1
phon_R01.S04_4	84	1	1	1	1
phon_R01.S04_5	14	1	1	1	1
phon_R01.S04_6	16	1	1	2	2
phon_R01.S05_1	27	1	1	2	2
phon_R01.S05_2	26	1	1	2	2
phon_R01.S05_3	26	1	1	2	2
phon_R01.S05_4	46	1	1	2	2
phon_R01.S05_5	46	1	1	2	2
phon_R01.S05_6	46	1	1	2	2
phon_R01.S07_1	31	0	1	1	1
phon_R01.S07_2	31	0	1	1	1
phon_R01.S07_3	31	0	1	1	1
phon_R01.S07_4	31	0	1	1	1
phon_R01.S07_5	31	0	1	1	1
phon_R01.S07_6	32	0	1	1	1

Tabla 8: Grupos asignados para los registros de voz pertenecientes a 5 personas, una de ellas no padece la EP

Como vemos en la tabla y en la información guardada en el archivo *G01.txt*, la red no parece ser demasiado acertada, ya que cuando se crean sólo 2 *clusters*, se agrupan una gran cantidad de sanos y enfermos en el mismo grupo. Esta información no es del todo clara al no tener una referencia sobre qué grupo está siendo asignado a los enfermos o a los sanos. Para tener una mejor perspectiva sobre cómo ha sido de preciso el entrenamiento debemos de evaluar si se han asignado a una determinada región ciertos registros pertenecientes a sanos o a enfermos. Para ello se ha implementado en Python una matriz sobre los datos del entrenamiento, filtrando sólo por sanos, y después sólo por enfermos. Las dos siguientes figuras (36 y 37) se muestran dichas matrices seguidas para poder comparar la distribución de los datos. Cada cuadrado representa la neurona en la que se encuentran los datos, conteniendo cada neurona el número de personas (sanas o enfermas, dependiendo de la figura), mostrándose con colores con tonos más oscuros según la cantidad de personas en la neurona. Todas estas figuras mencionadas serán de gran utilidad para la extracción de conclusiones para la detección de biomarcadores relacionados con la Enfermedad del Parkinson. Por ello también se realizarán gráficos sobre los pesos de las *features*, pudiendo observar la relación (y la no relación) entre los pesos de las características sobre el posicionamiento en la red de los registros sanos y enfermos. Todas estas imágenes se encuentran en el Anexo 1,

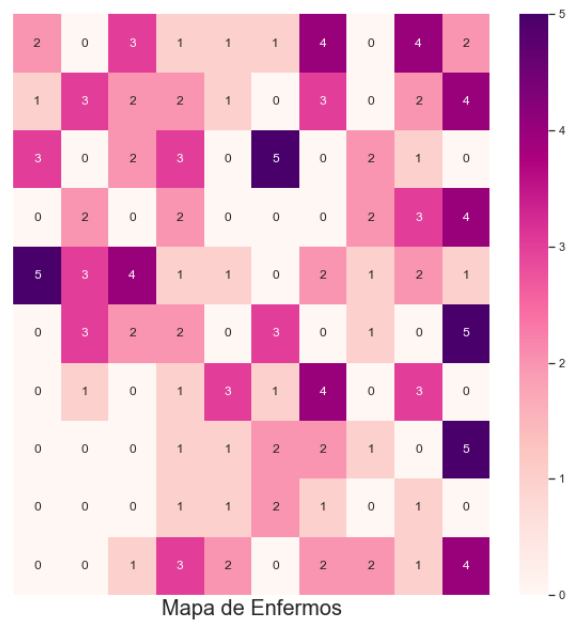


Figura 36: Matriz con la distribución de personas enfermas

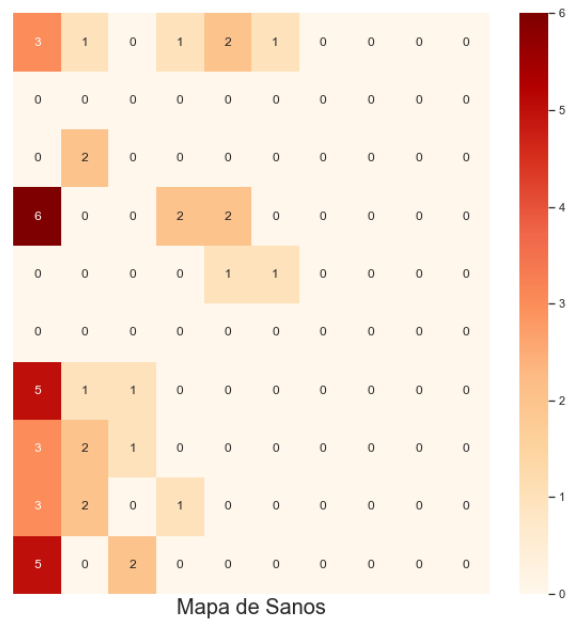


Figura 37: Matriz con la distribución de personas sanas

Como podemos observar, se aprecia una distribución casi uniforme para el mapa de enfermos, salvo un cuadrante en la parte inferior izquierda, donde la aparición de enfermos es casi nula. En ese mismo cuadrante encontramos una gran aparición de registros de personas sanas. Al parecer, la red ubica a las personas que no padecen la EP por esta región de la red, mientras que para las personas con la EP las distribuye por el resto de la red. Para poder visualizar esta información procedemos a juntar ambas matrices.

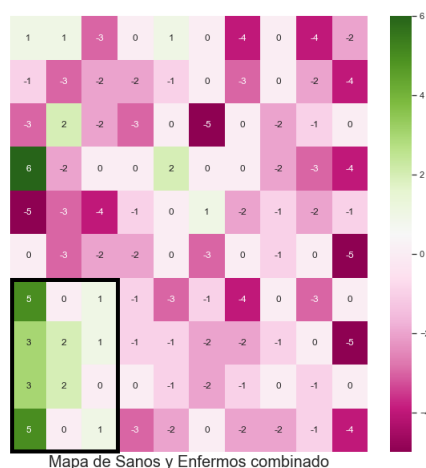


Figura 38: Diferencia entre la cantidad de sanos y enfermos por neurona (positivo y verde para sanos, negativo y rosa para enfermos)

Estas matrices nos aportan una visión diferente a las aportadas por el propio algoritmo, ya que dicho algoritmo generaba los grupos por la parte inferior derecha de la red, mientras que con estos gráficos podemos ver como realmente los datos se distribuyen en áreas diferentes, encontrándose los datos de personas sanas en la esquina inferior izquierda. Los datos restantes que no se encuentran en esta región de 3x4 neuronas se distribuyen sobretodo por la misma región del eje X, pero situado en el cuadrante superior, encontrándose un total de 23 registros fuera de la región de 3x4 neuronas. esto supone un 52 % de los registros en dicha región. Respecto al mapa de personas enfermas con la EP (Figura 36), se puede observar como se distribuyen por la mayoría de la red, siendo mucho más sencillo detectar una persona enferma que una sana utilizando redes SOM. Podemos observar que si realizamos la misma división que la planteada para el mapa de sanos, la precisión en este caso es prácticamente un 100 % (alrededor del 98,63 %), al sólo encontrarse 2 registros en esta región de 3x4. Esta precisión no es suficiente para poder determinar que este modelo haga una buena distribución de datos, pero es muy importante el estudio de éste, ya que este procedimiento será el que se seguirá con el resto de grupos de características, pudiendo comparar, en la recta final del proyecto, todos los modelos y grupos de características creados. Este grupo nos servirá para tener una medida base para el resto de modelos. Normalmente, se suele utilizar el modelo *ZeroR* como base de precisión para los modelos basados en *Machine Learning*, que se basa en predecir los datos futuros como la moda de los datos de entrenamientos. Para nuestro caso, esto no es posible por varias razones. Entre otras se encuentran que el modelo no realiza una predicción como tal, ya que proyecta la distribución de los datos en una imagen, quedando la predicción a libre interpretación (aunque es cierto que la mayoría de librerías también dotan de herramientas de predicción, como la división de *clusters* vista anteriormente). Otro punto en contra del uso de *ZeroR* en nuestro caso, es debido a la aparición poco realista del ratio de personas sanas respecto a personas enfermas, ya que como hemos dicho anteriormente, la EP la padece una parte de la población, pero no con un ratio tan alto. En las imágenes que tratan los pesos de cada característica de la red, podemos observar como existen pesos de características que se distribuyen con bastante relación a la distribución de personas sanas y enfermas con la EP mostradas en las Figuras 36, 37 y consecuentemente en la Figura 38. Esto se puede observar en características como *spread1*, *spread2*, *PPE* y ligeramente en

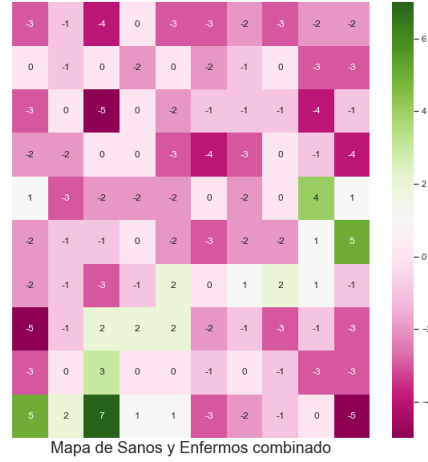


Figura 40: Diferencia entre la cantidad de sanos y enfermos por neurona

Se puede observar como se forma una línea que empieza en la esquina inferior izquierda, que atraviesa toda la red, acabando en la parte central del borde derecho de la red. Esta línea representa las personas sanas, mientras que las personas enfermas se distribuyen por todo el resto de la red. Podemos apreciar el patrón repetido en el primer experimento, donde también se distribuían las personas enfermas por toda la red, mientras que las personas sanas se concentran en un área más pequeña. Esto puede ser debido a la diferencia de apariciones de personas enfermas respecto a personas sanas en los datos, ya que, como vimos anteriormente, existen más registros con personas enfermas que sanas, impactando directamente en la disposición de los datos en la red. También es interesante, al igual que en el primer experimento, mostrar la gráfica con la relación de los pesos de la neurona con mayor cantidad de personas enfermas y personas sanas.

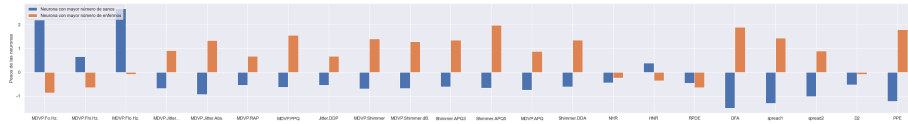


Figura 41: Pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos del experimento 2

7.4.3. Experimento 3: G02_SC_22_10x10

Para este modelo se ha modificado el ratio de aprendizaje solamente. Además, se ha realizado el paso contrario al grupo G02, aumentándolo, tanto su valor inicial, como su valor final, de (0.05, 0.01) a (0.1, 0.05).

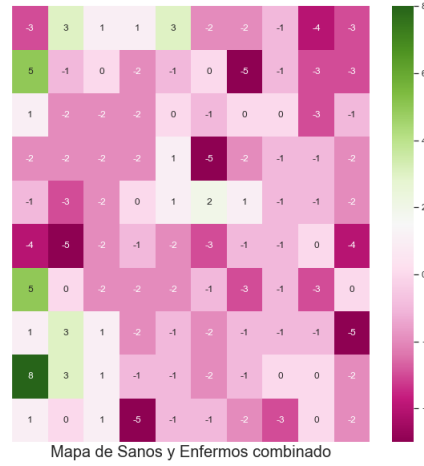


Figura 42: Diferencia entre la cantidad de sanos y enfermos por neurona

En el gráfico 42 se puede observar el parecido con el gráfico del experimento 1 (Figura 38), donde también se observa este patrón de distribución de personas sanas y enfermas. Esto puede dar a entender que aunque se aumente el ratio de aprendizaje de forma aislada, promedia los mismos resultados.



Figura 43: Pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos del experimento 3

En cuanto a la Figura 43, observamos el patrón anterior visto, donde las características referentes al *Shimmer* y *Jitter* tienen pesos muy parecidos entre ellas. Esto podría ser una de las causas de un mal entrenamiento, ya que se ponderan más ciertas características cuando sólo deberían existir una o dos características de esta naturaleza, tanto para el *Shimmer* como para el *Jitter*.

7.4.4. Experimento 4: G04_SC_22_10x10

Para esta red se ha aumentado solamente el número de *Epochs*, alterando la cantidad de los mismos, de la misma manera que en el G01, aumentando su cantidad (de 1000 a 2000).

7.4.5. Experimento 5: G05_SC_22_10x10

En este modelo se ha elegido reducir el tamaño inicial del radio de vecindad de la red SOM (en vez de cubrir 2/3 del tamaño de la red, empieza por 1/2). Con esto se consigue que se especialicen ciertas zonas en datos más concretos de una forma más rápida que con los valores por defecto.

7.4.6. Experimento 6: G06_SC_22_10x10

En este modelo se ha modificado el ratio de aprendizaje, aumentando el valor inicial, y reduciendo el valor final, intentando buscar un ratio de aprendizaje óptimo para la red. Este ratio empieza siendo 0.2 y en el último *Epoch* acaba siendo 0.005, mientras que en los valores por defecto empiezan por 0.05 y acaba siendo 0.01, como dijimos anteriormente.

7.4.7. Experimento 7: G07_SC_22_8x8

Este modelo es idéntico al primer experimento, distinguiéndose únicamente en el tamaño de la red de neuronas, siendo en este caso una red de 8x8 neuronas, por lo que en este caso también se mantienen los valores por defecto.

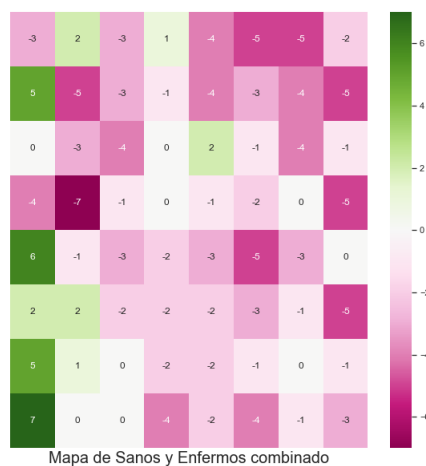


Figura 44: Diferencia entre la cantidad de sanos y enfermos por neurona

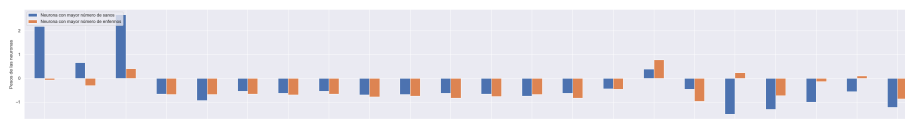


Figura 45: Pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos del experimento

7.4.8. Experimento 8: G08_SC_12_8x8

Para este experimento se ha cambiado el conjunto de características según las observaciones en los experimentos anteriores. En este caso, se parte de los gráficos mostrados en el experimento 7 (Figuras 44 y 45), donde se puede observar como, al crear una red de dimensiones inferiores al resto de experimentos, comienzan a igualarse pesos de características que en experimentos anteriores parecían dar indicaciones de dicha tendencia a la igualdad. Por ello, se ha decidido probar a entrenar una red de neuronas sin dichas características, intentando buscar un mejor resultado al eliminar características con pesos iguales. Pese a eliminar estas características, se ha decidido mantener una característica por naturaleza, es decir, una característica para el *Jitter* (en concreto *Jitter.Abs* al ser la más distante con su homónima en la otra neurona), y una característica para el *Shimmer* (en concreto *Shimmer.DDA* al ser también la más distante con su homónima). El resto de características que forman el G08 pertenecen a las tres primeras características que aparecen en el gráfico 45, anteriormente mencionado para la justificación del tamaño de la red, y las últimas siete características de este mismo gráfico. Estas características se han mantenido debido a la gran relevancia que parecen tener a la hora de escoger a personas sanas o enfermas, ya que son las que ofrecen pesos más distantes entre ellos mismos con sus homónimos en la neurona contraria.

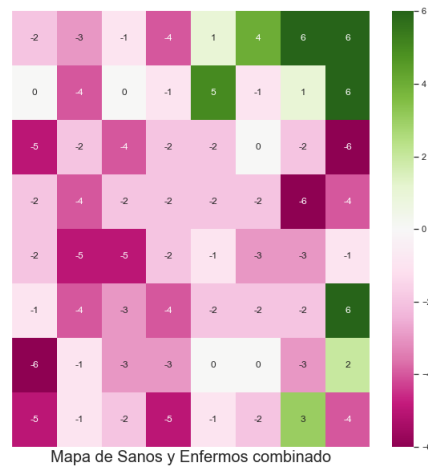


Figura 46: Diferencia entre la cantidad de sanos y enfermos por neurona

Pese a eliminar prácticamente la mitad de características del *DataSet* se observan resultados parecidos con los experimentos en los que sí se utilizaban dichas características, pudiendo indicar que, efectivamente, no permitían a la red un mejor entrenamiento.

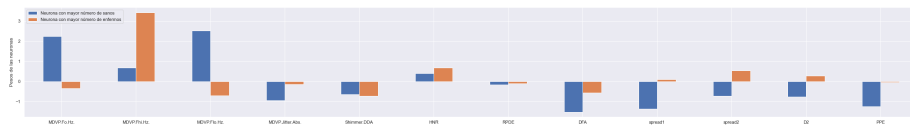


Figura 47: Pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos del experimento 8

En cuanto a los pesos de las características, se siguen obteniendo resultados similares a los experimentos anteriores, donde los pesos de las características distan bastante entre sus homónimos en la neurona contraria.

En la siguiente Figura se puede ver el conjunto de las gráficas sobre los pesos de las neuronas con mayor cantidad de enfermos y sanos, que igualmente se encuentra en el Anexo 1.

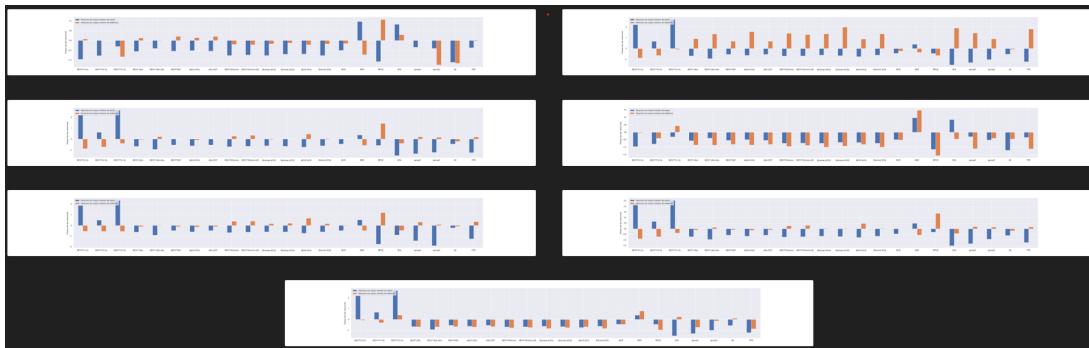


Figura 48: Conjunto de gráficas sobre los pesos de las neuronas con mayor cantidad de enfermos y mayor cantidad de sanos por modelo

8. Estudio y análisis

Una vez realizado una variedad de modelos, junto a las representaciones gráficas de cada uno de ellos, vamos a proceder a extraer información valiosa sobre ellos. Para ello, se harán referencias tanto a las imágenes expuestas en esta memoria, como a las imágenes contenidas en los diferentes Anexos. Estas conclusiones son vitales para el posible avance científico en el diagnóstico de la enfermedad de Parkinson, aunque los avances expuestos en este proyecto tendrán que ser extendidos y probados, debido a posibles errores externos al alcance del proyecto. Estos avances serán contemplados como proyectos futuros, ya que durante la creación y progresión del proyecto se han ido descubriendo distintos aspectos a mejorar, fuera del alcance de este proyecto.

9. Conclusiones

En este apartado se parte de las imágenes, datos y estadísticas obtenidos en apartados anteriores sobre los modelos creados. Gracias a los modelos creados con el G01 de características (modelos G01 - G07) se ha podido observar una relación directa entre algunas características del *DataSet* usado en el proyecto y el diagnóstico negativo de la enfermedad del Parkinson. Esto ocurre con las características *MDVP:Fhi(Hz)*, *MDVP:Flo(Hz)*, *spread1* y *spread2* especialmente. En otras características se puede apreciar un patrón parecido, aunque no se podría asegurar con claridad dicha correlación. Estas características corresponden al *PPE*, *RPDE* y *DFA*. Con el resto de características no se puede extraer ninguna relación, ya que no se corresponden a ningún patrón claro correspondiente a uno de los grupos (sanos o enfermos), o por que los pesos son muy pronunciados en prácticamente toda la red, como el caso de *HNR*, donde todas las neuronas tienen pesos muy parecidos salvo la esquina inferior derecha del primero modelo, y en el resto de modelos variando la posición pero con el mismo patrón, no representando dicha área ningún grupo de los antes mencionados (sanos o enfermos). Con estas imágenes también se puede ver cómo las *features* con la misma procedencia, como son las relativas al *Jitter* y *Shimmer*, cuentan con mapas relativamente parecidos, alterando a la red de neuronas de la misma manera, suponiendo la repetición de pesos y por ende, mayor influencia sobre los datos.

También observamos que en los pesos de las neuronas con mayor cantidad de personas sanas y personas enfermas, encontramos que en todos los modelos con el primer grupo de características (G01 - G07) los pesos de las neuronas se distancian visiblemente en las características que parecían indicar patrones separando las regiones de personas sanas y personas enfermas, anteriormente mencionadas. Se puede observar que según se reduce el tamaño de la red de neuronas, más cercanos son los pesos de las características que parecían no aportar información útil, como *MDVP:Shimmer* (y todas las de su naturaleza), *NHR* y *HNR*, mientras *spread1* y *spread2* se encuentran siempre distanciadas de su homónima en la otra neurona. Vemos esto mismo en las características de *MDVP:Fhi(Hz)* y *MDVP:Flo(Hz)*, como ya mencionamos anteriormente observando los mapas de los pesos de toda la red. En especial, se puede observar que según se reduce con el tamaño de la red, pero no con los valores del ratio de aprendizaje, ya que observamos como en los que se han reducido y aumentado, han seguido manteniendo cierta distancia entre las características que se han acabado solapando en modelos posteriores (como el G07 o G04). También se puede concluir que el número de *Epochs* influye ligeramente en el acercamiento de los pesos de dichas características.

9.1. Implicaciones Sociales y Medioambientales

10. Bibliografía

Referencias

- [1] ¿Cuáles son los tipos de inteligencia artificial que existen? 2022. URL: <https://www.inesdi.com/blog/tipos-de-inteligencia-artificial/>.
- [2] 'Machine learning': ¿qué es y cómo funciona? 2019. URL: <https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/>.
- [3] Ayodeji A Akinduko y Evgeny M Mirkes. «Initialization of self-organizing maps: principal components versus random initialization. A case study». En: *arXiv preprint arXiv:1210.5873* (2012).
- [4] *Alteraciones del Habla: DISARTRIA*. 2021. URL: <https://www.clinicasanvicente.es/servicios/alteraciones-del-habla-disartria/>.
- [5] TH Benke et al. «Repetitive speech phenomena in Parkinson's disease». En: *Journal of Neurology, Neurosurgery & Psychiatry* 69.3 (2000), págs. 319-324.
- [6] JoséManuel Calvo-Gómez et al. «Validación de una escala para evaluación de síntomas colaterales extrapiramidales de Simpson-Angus». En: *Revista de Salud Pública* 8.1 (2006), págs. 74-87.
- [7] Iria Carballo-Carbajal et al. «Brain tyrosinase overexpression implicates age-dependent neuromelanin production in Parkinson's disease pathogenesis». En: *Nature communications* 10.1 (2019), pág. 973.
- [8] Mario Carranza. «Fenómenos de interferencia fónica relacionados con el fonema /w/ en la interlengua de estudiantes japoneses de español como lengua extranjera». En: *Estudios Lingüísticos Hispánicos* 23 (2008), págs. 1-22.
- [9] K Ray Chaudhuri et al. «King's Parkinson's disease pain scale, the first scale for pain in PD: an international validation». En: *Movement Disorders* 30.12 (2015), págs. 1623-1631.
- [10] Ignacio Cobeta, Faustino Núñez y Secundino Fernández. *Patología de la voz*. Marge books, 2013.
- [11] R Cano de la Cuerda et al. «Escala de valoración y tratamiento fisioterápico en la enfermedad de Parkinson». En: *Fisioterapia* 26.4 (2004), págs. 201-210.
- [12] Miet De Letter, Patrick Santens y John Van Borsel. «The effects of levodopa on word intelligibility in Parkinson's disease». En: *Journal of communication disorders* 38.3 (2005), págs. 187-196.
- [13] Marcos Domínguez. «Díez Tejedor: "Necesitamos una redistribución de los neurólogos en España"». En: *Redacción médica* (2018). URL: <https://www.redaccionmedica.com/secciones/neurologia/diez-tejedor-necesitamos-una-redistribucion-de-los-neurologos-en-espana--8174>.
- [14] E Dorsey et al. «The emerging evidence of the Parkinson pandemic». En: *Journal of Parkinson's disease* 8.s1 (2018), S3-S8.
- [15] Dr. Jon Infante Ceberio Dra. María Álvarez Saúco. *Causas del Parkinson*. URL: <https://conoceelparkinson.org/causas-del-parkinson/>.
- [16] *Enfermedad de Parkinson*. Mar. de 2023. URL: <https://www.mayoclinic.org/es-es/diseases-conditions/parkinsons-disease/symptoms-causes/syc-20376055>.
- [17] *Enfermedad del Parkinson*. Mar. de 2019. URL: <https://residenciamirasol.com/enfermedad-del-parkinson/>.
- [18] KA Flowers, C Robertson y MR Sheridan. «Some characteristics of word fluency in Parkinson's disease». En: *Journal of Neurolinguistics* 9.1 (1995), págs. 33-46.
- [19] Carlos Galván Duque, Ricardo Zavala-Yoe y Ricardo Ramirez-Mendoza. «DISEÑO, MODELADO Y CONTROL POR SIMULACIÓN NUMÉRICA DE ARTICULACIÓN MAGNETOREOLÓGICA DE RODILLA PARA ÓRTESIS DE MIEMBRO INFERIOR». Tesis doct. Mayo de 2015.
- [20] Leonor García. «Málaga tiene apenas la mitad de los neurólogos que debería por su población». En: *Málaga hoy* (2022). URL: https://www.malahoy.es/malaga/Malaga-neurologos-poblacion%5C_0%5C_1667534644.html.
- [21] Núria Gavalda. ¿Qué es un Oscilograma? URL: <https://nuriagavalda.com/que-es-un-oscilograma/>.

- [22] Ligdi González. *¿Qué son las Redes Neuronales Artificiales?* 2021. URL: <https://aprendeia.com/que-son-las-redes-neuronales-artificiales/>.
- [23] Andrew Graham. «HyperPhysics». En: *The Physics Teacher* 40.5 (2002), págs. 318-318.
- [24] *How is Parkinson's diagnosed?* 2018. URL: <https://www.clinicasanvicente.es/servicios/alteraciones-del-habla-disartria/>.
- [25] Andrew J Hughes et al. «Accuracy of clinical diagnosis of idiopathic Parkinson's disease: a clinico-pathological study of 100 cases.» En: *Journal of neurology, neurosurgery & psychiatry* 55.3 (1992), págs. 181-184.
- [26] *Inarmonía*. URL: <https://ears2.dmu.ac.uk/es/encyclopedia/inarmonia>.
- [27] karla_ponce7. *Aparato fonador*. URL: <https://quizlet.com/pr/261102992/aparato-fonador-diagram/>.
- [28] T. Kohonen. «The self-organizing map». En: *Proceedings of the IEEE* 78.9 (1990), págs. 1464-1480. DOI: 10.1109/5.58325.
- [29] Juan Diego Hernández Lalinde et al. «Sobre el uso adecuado del coeficiente de correlación de Pearson: definición, propiedades y suposiciones». En: *Archivos venezolanos de Farmacología y Terapéutica* 37.5 (2018), págs. 587-595.
- [30] Max Little et al. «Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection». En: *Nature Precedings* (2007), págs. 1-1.
- [31] Max Little et al. «Suitability of dysphonia measurements for telemonitoring of Parkinson's disease». En: *Nature Precedings* (2008), págs. 1-1.
- [32] T Soni Madhulatha. «An overview on clustering methods». En: *arXiv preprint arXiv:1205.1117* (2012).
- [33] P.Martínez Martín y D.García Urrua. *Escala de la marcha para Enfermedad de Parkinson - (Versión 2.0)*. 1996. URL: <https://getm.sen.es/profesionales/escalas-de-valoracion/26-getm/escalas-de-valoracion/73-escala-de-la-marcha-para-enfermedad-de-parkinson-version-2-0>.
- [34] Ana Pérez Menéndez. *El número de afectados por la enfermedad de Parkinson se duplicará en 20 años y se triplicará en 2050*. 2018. URL: <https://www.sen.es/saladeprensa/pdf/Link238.pdf>.
- [35] N Nallamuthu. «Handling Imbalanced Data-Machine Learning Computer Vision and NLP». En: *Data Science Blogathon* (2020).
- [36] F. Pedregosa et al. «Scikit-learn: Machine Learning in Python». En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.
- [37] Astrid-Helene Ravn, Jacob P Thyssen y Alexander Egeberg. «Skin disorders in Parkinson's disease: potential biomarkers and risk factors». En: *Clinical, cosmetic and investigational dermatology* (2017), págs. 87-92.
- [38] Alejandro Rodriguez-Ruiz et al. «Stand-alone artificial intelligence for breast cancer detection in mammography: comparison with 101 radiologists». En: *JNCI: Journal of the National Cancer Institute* 111.9 (2019), págs. 916-922.
- [39] Betul Erdogan Sakar et al. «Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings». En: *IEEE Journal of Biomedical and Health Informatics* 17.4 (2013), págs. 828-834.
- [40] M.D. et al. Sandhya Pruthi. *Enfermedad de Parkinson*. 2023. URL: <https://www.mayoclinic.org/es-es/diseases-conditions/parkinsons-disease/diagnosis-treatment/drc-20376062>.
- [41] M.D. et al. Sandhya Pruthi. *Tomografía computarizada por emisión de fotón único*. 2022. URL: <https://www.mayoclinic.org/es-es/tests-procedures/spect-scan/about/pac-20384925>.
- [42] sys-admin. *Qué es bradicinesia*. 2018.
- [43] Begoña Torres y Ferran Gimeno. *Anatomía de la Voz*. Paidotribo, 2008.
- [44] JF Vázquez-Costa et al. «Análisis del trayecto y retraso diagnóstico de los pacientes con esclerosis lateral amiotrófica en la Comunidad Valenciana». En: *Neurología* 36.7 (2021), págs. 504-513.

- [45] Ron Wehrens y Lutgarde M. C. Buydens. «Self- and Super-Organizing Maps in R: The kohonen Package». En: *Journal of Statistical Software* 21.5 (2007), págs. 1-19. DOI: 10.18637/jss.v021.i05.
- [46] Ron Wehrens y Johannes Kruisselbrink. «Flexible Self-Organizing Maps in kohonen 3.0». En: *Journal of Statistical Software* 87.7 (2018), págs. 1-18. DOI: 10.18637/jss.v087.i07.
- [47] *What is parkinson's?* Feb. de 2018. URL: <https://www.parkinsonseurope.org/about-parkinsons/what-is-parkinsons/>.

11. Anexos

1. Se adjunta en versión .ZIP el conjunto de gráficas y estadísticas creadas y estudiadas en este trabajo.