UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

título del TFG

Presentado por Adrián Arnaiz Rodríguez
en Universidad de Burgos — 13 de junio
de 2019
Tutor: nombre tutor

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Adrián Arnaiz Rodríguez, con DNI 71306880A, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 13 de junio de 2019

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. nombre tutor	D. nombre co-tutor

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice de figuras

Índice de tablas

Introducción

1.1. Introducción al proyecto

Se hará una introducción al proyecto completo

1.2. Estructura de la memoria

Se explicará como está estructurada la memoria del proyecto.

Objetivos del proyecto

2.1. Objetivos generales

- Investigar y condensar el estado del arte de la investigación sobre la detección del Parkison a través de la voz resumiendo los artículos científicos más importantes, identificando las tecnologías, herramientas y procesos actuales utilizadas en ellos, etc.

- Recopilación de bases de datos adecuadas para la investigación, tanto para uso en este proyecto como para su uso en proyectos posteriores, cerciorando que son *datasets* de audios correctos para las tareas necesitadas (i.e. audios etiquetados).

- Realización de un estudio comparativo en cuanto a la utilización de diferentes modelos de clasificación y conjuntos de características extraídas de los audios. Se compararán resultados, tanto entre los diferentes experimentos que nosotros realicemos, como entre nuestros mejores experimentos y resultados científicos de anteriores experimentos (publicados en artículos científicos).

- Aportación de una nueva perspectiva a este campo de investigación: extracción de las características de los audios mediante *Deep Learning*.

- Finalmente, utilizar todo lo descrito anteriormente para realizar una aplicación web la cual permita la monitorización de la detección de la enfermedad del Parkinson a través de la voz. La aplicación será capaz de discernir, mediante un clasificador, si la persona de un audio subido a la aplicación web tiene Parkinson o no.

2.2. Objetivos técnicos

- Desarrollar un algoritmo, cuya implementación en Python permita la extracción de características de los audios de manera adecuada (envolver la extracción de características que realizan Disvoice y VGGish en clases).
- Desarrollar una aplicación web para la étapa de explotación (más detalles cuando lo realicemos más adelante) la cual deberá estar alojada en un servidor Flask dentro de un contenedor Docker. ¿Lo haremos?, ¿De esta manera?
- Utilizar las herramientas más adecuadas para la realización del estudio comparativo: valoraremos Python y sus bibliotecas, Weka...
- Utilizar un sistema de control de versiones Git utilizando para ello la plataforma Github. Utilizar un cliente Git para el trabajo local para lo cual utilizaremos TortoiseGit.
- Utilizar la metodología Scrum en la elaboración del proyecto y concretamente la herramienta ZenHub (como extensión de Github) para la ayuda en la gestión de proyectos.

Conceptos teóricos

Este proyecto abordará diferentes temas dentro del campo conocido como *Data Mining*. A su vez se tratan temas sobre el análisis del discurso desde una perspectiva física. Se explicarán conceptos dentro de ambos campos para la correcta comprensión del proyecto.

3.1. Minería de datos y aprendizaje automático

La **minería de datos** es un campo del conocimiento dentro de las ciencias de computación cuyo objetivo general es analizar grandes volúmenes de datos para extraer conocimiento de ellos. Se basa en el concepto de que la sociedad, sobre todo actualmente, produce una gran cantidad de datos y de diferente origen. Sin embargo, extraer conocimiento de los datos no es un proceso tan trivial. No es posible sacar la información de los datos en crudo. Por ello se necesitan diferentes métodos y técnicas que nos permitan hacer esa tarea, como son las técnicas de **aprendizaje automático** [26]. Este proceso de divide en varias fases.

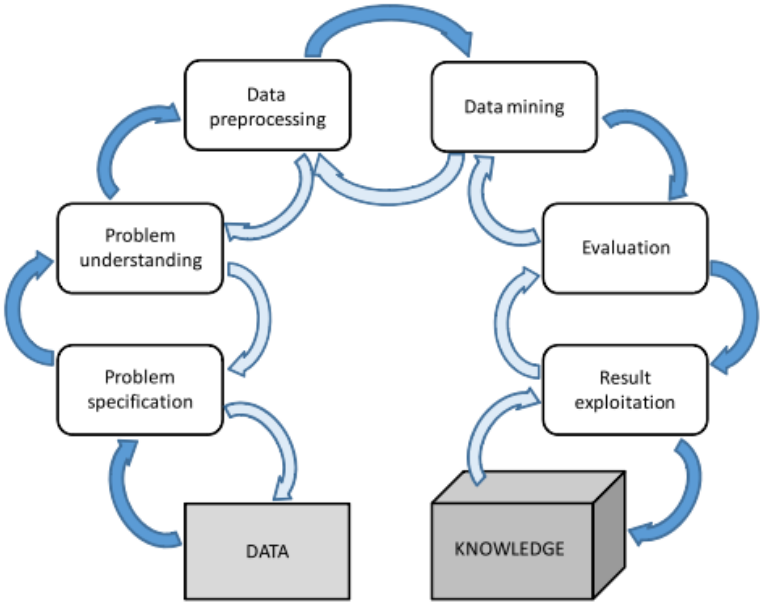

Figura 3.1: Fases en la extracción de conocimiento de un conjunto de datos. [8]

Podemos hablar de que la minería de datos es el análisis de un conjunto grande de datos para la obtención de conocimiento de ellos, ya sea relaciones entre esos datos o patrones ocultos, que nos permita obtener una ventaja competitiva o un conocimiento novedoso. Esto está relacionado con el proyecto en cuanto a que queremos obtener conocimiento de la relación voz-enfermedad del Parkinson a través un gran conjunto de audios. A continuación, definiremos varios conceptos dentro de este campo que son manejados en el proyecto.

Pre-procesamiento de datos

Como hemos comentado anteriormente, los datos no pueden ser tratados de manera directa, necesitan ser tratados en una etapa anterior al descubrimiento de información: pre-procesamiento. Esto es indispensable en la fase de pre-procesamiento, transformar los datos en bruto a otro conjunto de datos que pueda servir para ser procesado por los algoritmos necesarios. Específicamente en nuestro caso, no podemos obtener conocimiento de el conjunto de audios en bruto, estos audios deberán ser tratados para extraer un conjunto de características numéricas de cada uno de ellos y poder seguir avanzando a partir de ese punto. El conjunto de características extraídas

de los audios será en su mayoría propiedades físicas (3.6) o características extraídas con bibliotecas de *Deep Learning*. Como nuestro proyecto se basará en aprendizaje supervisado, también se llevará a cabo en esta etapa el etiquetado de los datos.

Otra parte importante, aunque no indispensable, dentro de esta etapa es la fase de selección de características. Cuando tenemos un gran número de atributos para cada instancia puede ser recomendable reducir su dimensionalidad. Una manera de reducir su dimensionalidad es a través de la selección de atributos manual, mediante un análisis estadístico. Un ejemplo de su uso puede verse en [23], donde varios de ellos son utilizados para definir un subconjunto de 10 características de cada audio dentro de las 132 medidas que se obtenían inicialmente en cada uno de ellos. En nuestro caso, no se realiza ese proceso manualmente. Por un lado, hemos realizado algunos experimentos donde hemos proporcionado todos los atributos de cada audio al algoritmo de clasificación y trasladando a éste la tarea de discernir cuales serán más importantes. Por otro, hemos realizado experimentos donde, en el proceso, se utilizan diferentes métodos selectores de atributos como *Select K Best* o *Variance Treshold* CITAR SELECTORES, REFERENCIAR A SU APARTADO 3.4 (ver sección 3.1). [8]

Aprendizaje automático

Es el campo de las ciencias de la computación enfocado a que los dispositivos *aprendan* por ellos mismos sin haber sido explícitamente programados para ello. Esta es una rama de la inteligencia artificial. Tras obtener un conjunto válido de características se aplican algoritmos de aprendizaje automático. Estos algoritmos son los encargados de detectar patrones en los datos. Estos algoritmos tienen un fuerte componente matemático y estadístico ya que, a través de métodos de estos campos del conocimientos, se extrae la información de los datos. Se divide en:

- **Aprendizaje supervisado**: para cada instancia del conjunto de datos tendremos tanto la entrada como la salida deseada. El objetivo será predecir la salida correcta para una nueva entrada. En nuestro caso la entrada serían las características de los audios y la salida si la persona de ese audio tiene parkinson (clasificación) o en qué nivel lo tiene (regresión) [1].

[1] Estos ejemplos ilustran la diferencia entre clasificación, donde hay que predecir una etiqueta de entre un conjunto finito de etiquetas posibles, y regresión, donde el valor que se predice es una magnitud continua de un conjunto con un número de valores

- **Aprendizaje no supervisado**: únicamente se tienen los datos de entrada y no se sabe nada acerca de su salida o clase. Habitualmente estos algoritmos son utilizados para la detección de clases que permitan separar los datos en diferentes grupos. Nosotros no utilizaremos este tipo en nuestro proyecto.

Generalización

El objetivo de los algoritmos de aprendizaje supervisado no es lograr dar salidas correctas para los ejemplos de datos que tenemos, sino lograr clasificar o dar la salida correcta para un nuevo ejemplo que nos llegue. Se puede decir que el objetivo del aprendizaje automático es en realidad es obtener conocimiento a partir de un gran conjunto de datos el cual sea extrapolable para todos los datos dentro de esa misma situación (de ese mismo problema). El problema de ajustarse demasiado a los datos de entrenamiento perdiendo así capacidad de generalización se llama sobreajuste. Esto ocurre cuando se crean modelos demasiado complejos que se ajustan a los datos de entrenamiento al 100 %. Al ocurrir sobreajuste, corremos el riesgo de clasificar mal nuevos datos. Si además este sobreajuste a los datos de entrenamiento se combina con la aparición de datos ruidosos en nuestro dataset, se perderá aún más capacidad de generalización. Por ello es importante tener métodos de evaluación de modelos que nos permitan medir cuánto de bueno es nuestro modelo a la hora de generalizar. Aquí aparece el concepto **Cross-Validation**.

K-fold Cross Validation [18]

La validación cruzada es un método para la evaluación y comparación de algoritmos. Se basa en descomponer el conjunto de datos en 2 subconjuntos, entrenamiento y test, con el objetivo de evaluar que tal generaliza nuestro modelo. Sin embargo, de esta manera estamos perdiendo información de nuestros datos, ya que el conjunto que utilizamos para test nunca es usado para entrenar. La solución a este problema es la validación cruzada k-fold.

En **k-fold cross-validation** el conjunto entero de datos se divide en k conjuntos del mismo tamaño cada uno. A partir de ahí se itera sobre los subconjuntos k veces utilizando cada vez $k-1$ conjuntos para entrenar y 1 conjunto de test. Para cada una de las iteraciones se guarda la performance y finalmente se hace la media para obtener una performance total. El número idóneo de k es 10 según [18] y así es utilizado el 10-fold en [15].

---

potencialmente infinito.

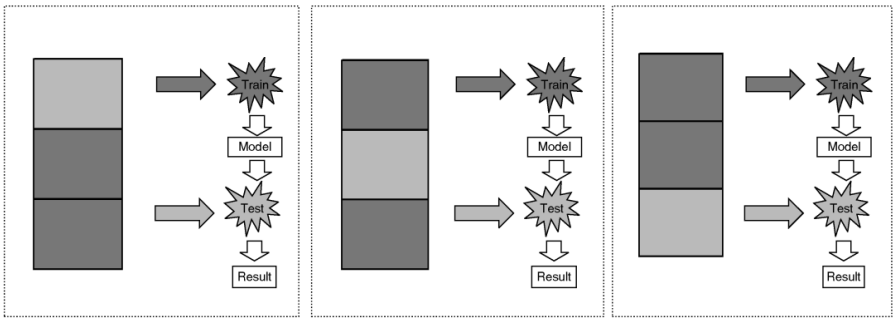

Figura 3.2: Ejemplo 3-fold cross-validation extraído de [18].

Cabe destacar que [18] explica que para *k-fold cross-validation* se debe realizar estratificación. **Estratificación** es el proceso por el cual se asegura que una de las k particiones de los datos total es una buena representación de los datos totales. Por ejemplo, en nuestro caso tenemos 50 % de personas con Parkinson y 50 % de personas sanas. Esto significa que para estratificar de manera correcta, cada partición deberá tener la mitad de instancias de cada clase. Cada partición k deberá tener aproximadamente la misma proporción de clases que el conjunto total.

Nested Cross Validation

DEFINIR.

Algoritmos de aprendizaje automático

A continuación explicaremos varios algoritmos de los utilizados en este proyecto.

SVM [6]

Este algoritmo de aprendizaje supervisado es llamado como máquinas de vector de soporte (SVM). Es un clasificador lineal basado en el concepto de margen máximo. Se utiliza tanto para regresión como para clasificación. Para clasificación su principio fundamental es separar dos clases en el conjunto de datos utilizando un hiperplano. El hiperplano utilizado como separador de las clases será el que maximice el margen entre las clases. Como sabemos, los problemas del mundo real tienen complicaciones como que no sean linealmente separables, que las clases estén solapadas, que haya más de 2 clases de datos, etc.

Para solucionar el problema de la existencia de más de 2 clases, se utiliza un SVM por cada una de las clases. Sin embargo, para el problemas de separar dos clases que no son linealmente separables, utilizamos el **kernel trick**. Esta característica del algoritmo es importante para poder separar este tipo de datos. Estas funciones kernel son utilizadas para aumentar la dimensionalidad de los datos de entrada, es decir, transformar el espacio de entrada. De esta manera dividiendo linealmente el espacio transformado con el hiperplano pueden resolverse el problema de la separación lineal. Existen diferentes ejemplos de funciones kernel, entre las más habituales están el kernel, polinomial, funciones de base radial, sigmoide y kernel Gaussiano que es utilizado en [23] y [15].

Random Forest [3]

Es un método de **ensemble** (métodos combinados) basado en los árboles de predicción, la combinación de la selección aleatoria de atributos y el *bagging*. El proceso es el siguiente: se construirán de manera independiente un conjunto de árboles, los cuales serán todos diferentes, debido a que están construidos con *bagging* y aleatoriedad de atributos, a la hora de predecir una nueva instancia, cada uno hace su predicción y, teniendo en cuenta la decisión de todos los árboles, llegamos a una decisión final conjunta. **Bagging** consiste en obtener varios subconjuntos de datos a partir de un único conjunto mediante remuestreo con reemplazamiento. Esto se hace escogiendo N elementos del conjunto inicial de manera aleatoria pudiendo coger el mismo ejemplo varias veces. Esto se acopla a la construcción de cada árbol de la siguiente manera:

1. A la hora escoger el atributo discriminante de un nodo, el mejor lo elegiremos de entre un subconjunto de todos los atributos. El mejor tamaño de este subconjunto es de tamaño $\sqrt{M}$ o $\log M$, siendo M el número de atributos.

2. Se realiza ese proceso para cada nodo del árbol.

A la hora de realizar la predicción del error se sigue el método **out of bag**. Consiste en recorrer las instancias de entrenamiento y predecir la clase de cada una únicamente con el conjunto de árboles que no han tenido esa instancia en su conjunto de entrenamiento. Se hace ese proceso para todas las instancias que tenemos y predecimos el error.

Algoritmos de selección de atributos

DEFINIR.

Grid Search, selección de parámetros

DEFINIR.

3.2. Deep Learning

Se describirá cuando se vea.

3.3. Conceptos estadísticos

Se describirán a continuación una serie de conceptos matemático-estadísticos utilizados tanto en la evaluación de clasificadores, como en la obtención de medidas de los audios.

ROC

Se describirá cuando se vea.

Medidas de distribución

Son diferentes medidas extraídas de cada medida física de los audios. Por ejemplo, de la amplitud de onda de un audio se sacan sus 4 funcionales correspondientes con sus momentos de distribución (media, desviación típica, coeficiente de asimetría, curtosis). Se obtienen debido a que caracterizan una muestra de tal manera que si dos distribuciones tienen los momentos iguales son iguales.

- **Media**: Es el resultado de la suma de todos los valores dividida entre el número de ellos. Se corresponde con el momento de la distribución de orden 1 respecto a la origen. m.
- **Desviación**: Mide la dispersión que tienen unos datos respecto a la media. Cuanto mayor sea este valor significará que los datos se encuentran en un rango amplio respecto a la media, mientras que si su valor en bajo significará que los valores se agrupan en un rango cercano a la media. Se corresponde con el momento de la distribución de orden 2 respecto a la media. d.

- **Oblicuidad o coeficiente de asimetría**: Mide la mayor o menor simetría de la distribución. Contra mayor sea el coeficiente de oblicuidad, mayor será simetría de los datos respecto a la media. Se corresponde con el momento de la distribución de orden 3 respecto a la media. sk.

- **Curtosis**: Mide la mayor o menor concentración de datos alrededor de la media. A un mayor valor de este coeficiente, se entiende que los valores están más agrupados en torno a la media y en valores alejados de ella, dejando los tramos intermedios con menor frecuencia. Se corresponde con el momento de la distribución de orden 4 respecto a la media. k.

Momento de la distribución de orden r respecto a la media:

$$m_r = \sum_{i=1}^{n} (x_i - \overline{x})^r P(n_i) \tag{3.1}$$

3.4. UPDRS

Unified Parkinson's disease rating scale [7], **UPDRS** es la escala universal con la que se mide el grado de severidad del Parkinson. Fue creada para dar un estándar a la evaluación de áreas específicas de la discapacidad. Esta escala evalúa 6 partes fundamentales: 1-Comportamiento, 2-Actividades de la vida diaria, **3-Examen motor** [21]; 4-Complicaciones de la terapia, 5-Escala de Hoehn & Yahr (Severidad) [10] y 6-Escala de Schwab y England (vida cotidiana).

UPDRS-III

Debido al objetivo de nuestro estudio, nos centraremos en la parte 3: examen motor [21]. Utiliza una escala de 0-4, donde 0 es la ausencia de discapacidad motora y 4 discapacidad motora severa. Esta parte trata diversos temas todos relacionados con la capacidad motora corporal. Mide entre otras cosas elementos como el habla, la expresión facial, el temblor o la agilidad. Todo esto afecta a la voz en cuanto se debe realizar un control correcto de los músculos glotales, labios, faringe y más para conseguir una pronunciación satisfactoria. La discapacidad motora produce a la hora de hablar volúmenes bajos, discurso monótono, articulación imprecisa [15]. El englobe de estas capacidades se denomina disartria hipocinética [9].

Hoehn & Yahr

La escala de Hoehn & Yahr [10] evalúa la enfermedad del Parkinson en una escala del 1 al 5. La enfermedad es más severa en función del aumento de la escala. El grado mínimo 1 se corresponde con que la enfermedad es exclusivamente unilateral, aumentando hasta el grado 5 en el que el paciente está en silla de ruedas o en cama, si no tiene ayuda.

3.5. Análisis del discurso

El análisis del discurso se realiza desde 3 prismas diferentes, que son la fonación, la articulación y la prosodia [19]. En nuestro proyecto, fijaremos estos grupos de características a extraer de cada tipo de audios.

Análisis de la fonación

La fonación aborda la vibración de las cuerdas vocales a la hora producir un sonido [19]. Desde el punto de vista clínico, este análisis está relacionado con la curvatura y el cierre incompleto de las cuerdas vocales a la hora de emitir un sonido [17]. Las medidas más típicas para el análisis de fonación son Jitter, Shimmer, APQ y PPQ [16]. Explicadas en la sección 3.6.

Análisis de la articulación

La articulación comprende la modificación de la posición, el estrés y la forma de los órganos y tejidos involucrados en la producción del habla [19]. Esto se manifiesta en déficits como, por ejemplo, una reducción de la amplitud y la velocidad de los movimientos articulatorios de los labios, la mandíbula y la lengua [20]. La capacidad de articulación es evaluada a través de la energía que se libera en las transiciones entre segmentos *voiced* → *unvoiced* (transición entre segmentos con voz y sin voz) y viceversa con medidas como MFCC o BBE, explicadas en 3.6. Se basa en que los pacientes de PD tienen dificultad para comenzar y para detener la vibración de las cuerdas vocales [16].

Análisis de la prosodia

La prosodia aborda temas como la variación del volumen, el tono y la sincronización para hablar de manera natural [19]. Se manifiesta con monotonía en el volumen y en el tono, cambios de rapidez en el habla y dificultades a la hora de expresar emociones a través del discurso [13]. Las

medidas más típicas para el análisis de prosodia son las relacionadas con la frecuencia fundamental, el contorno de la energía y la duración. Explicadas en la sección 3.6.

3.6. Características Físicas de la voz

Desde el punto de vista ingenieril y relacionado con el aprendizaje automático, todos los análisis anteriores se deben expresar de una manera numérica para poder crear vectores de características de un audio. Por ello, se extraen diferentes características numéricas de los audios que intentan expresar los análisis de fonación, articulación y prosodia de la manera más óptima posible. Otra dificultad importante desde el punto de vista ingenieril es la continuidad de los audios. Los audios cambian de manera continua en el tiempo, lo que complica el proceso de la extracción de características. Por ello, el audio se divide pequeños segmentos de tiempo en los que se calculará las características de manera estática para ese tramo. Las medidas físicas extraídas de los archivos de voz son de gran variedad y por ello en este apartado explicamos las más comunes y más usadas en los diferentes estudios del análisis del discurso.

Jitter

El concepto **Jitter** es usado para evaluar la variabilidad temporal durante el envío de señales digitales [24]. Concretamente, en el análisis del discurso, esta medida significa la variación temporal de la frecuencia fundamental del discurso [16]. N es el número de fragmentos, M_f la frecuencia máxima y $F_0(k)$ la amplitud de ese frame en concreto:

$$Jitter(\,\%) = \frac{100}{NM_f} \sum_{k=1}^{N} |F_0(k) - M_f| \tag{3.2}$$

Shimmer

El concepto **Shimmer** es usado, en el análisis del discurso, para medir la variación temporal de la amplitud del discurso [16]. N es el número de *frames*, M_a la amplitud máxima y $A(k)$ la amplitud de ese *frame* en concreto:

$$Shimmer(\,\%) = \frac{100}{NM_a} \sum_{k=1}^{N} |A(k) - M_a| \tag{3.3}$$

APQ y PPQ

APQ mide la variabilidad a largo plazo de la amplitud de la voz. Para calcularla, se suaviza mediante una media móvil de tamaño 11 y se calcula como la diferencia media absoluta entre la amplitud de un *frame* y las amplitudes promediadas sobre sus vecinos, dividida por la amplitud media.

La **PPQ** mide la variabilidad a largo plazo de la frecuencia fundamental y para calcularla se suaviza mediante una media móvil de tamaño 5. Se calcula como la diferencia promedio absoluta entre la frecuencia de cada cuadro y el promedio de sus vecinos, dividida por la frecuencia media. Ambas medidas se calculan de igual manera, APQ relativa a la amplitud y PPQ relativa a la frecuencia [16].

Mel Frequency Cepstral Coefficients

Los **Coeficientes Cepstrales en las Frecuencias de Mel** son 12 coeficientes para la representación del habla basados en la percepción auditiva humana. Con ellos podemos obtener la información más relevante de una porción de audio, obviando partes menos importantes como el ruido. El cálculo de estos coeficientes está basado en la transformada de Fourier y la transformada del coseno discreta [25]. Como hemos comentado anteriormente, nos sirve para medir la energía que se libera en las transiciones entre segmentos *voiced* $\rightarrow$ *unvoiced* y viceversa. Tiene un crecimiento logarítmico y se calcula según la siguiente función:

$$m = 1127,01048 \ln \frac{1+f}{700} \tag{3.4}$$

Bark Band Scale

Bark Band Scale es otro método de caracterización de la energía liberada en las transiciones de la voz. Da 24 coeficientes acorde a la escala de Bark [27]. Es una escala psicoacústica en la que se definen límites de frecuencias para definir la escala (i.e. grado 1 [100Hz-200Hz], grado 2 [200Hz-300Hz],..., grado 23 [12000-15500], grado 24 [15500- ∞]) Tiene un crecimiento logarítmico y se calcula según la siguiente función:

$$Bark(f) = 13 \arctan(0{,}00076 f) + 3{,}5 \arctan\left(\frac{f}{7500}\right)^2 \tag{3.5}$$

Medidas relacionadas con la frecuencia fundamental

Su objetivo es caracterizar los patrones de entonación de la voz. Por ello, se calcula el contorno de la frecuencia máxima y se calculan diferentes estadísticos sobre ella, como medias en Hz, desviación en Hz, máximos, etc [16].

Medidas relacionadas con la energía

Al igual que para el anterior caso, pero esta vez con la energía. Se calculan medidas estadísticas sobre el contorno de la energía: media en dB, desviación en dB, máximo o el coeficiente de regresión entre el contorno de energía y una regresión lineal [16].

Técnicas y herramientas

Explicaremos las herramientas utilizadas en nuestro proyecto.

4.1. Python

DEFINIR.

4.2. Anaconda Distribution

DEFINIR.

4.3. Jupyter Notebook IDE

DEFINIR.

4.4. Librerías de Python

Numpy

DEFINIR.

Scikit-learn

DEFINIR.

Scipy

DEFINIR.

Pandas

DEFINIR.

Pysptk

DEFINIR.

Sounddevice

DEFINIR.

Pydub

DEFINIR.

Disvoice

La biblioteca Disvoice[2] [16] es un conjunto de *scripts* de Python para la extracción de medidas del habla. Disvoice calcula medidas de articulación, de la fonación y de prosodia a partir de vocales sostenidas y expresiones verbales continuas, con el objetivo de evaluar las capacidades de comunicación de los pacientes con diferentes trastornos de la voz o trastornos neurodegenerativos como la enfermedad de Parkinson. Ha sido desarrolada por Juan Camilo Vásquez-Correa, el cual es co-autor de varios artículos con Juan Rafael Orozco-Arroyave como [16], y tiene licencia de software MIT. Cabe destacar que se han intercambiado dos correos electrónicos con JC Vásquez-Correa, en los que se nos explica tanto la utilización de los *scripts*, como la salida detallada de cada uno de ellos.

Contiene 3 *scripts* principales para la extracción de características de audios (fonación, articulación y prosodia), extrayendo por ejemplo de un audio hasta 488 medidas relacionadas con la articulación. Tiene una gran parte de su contenido dedicado a la visualización de los audios mediante diferentes métodos, característica que no es usada en nuestro proyecto.

[2] https://github.com/jcvasquezc/DisVoice

Esta biblioteca ha sido usada, como puede entenderse, para la etapa de extracción de características de los audios. Está escrita para ser usada como *scripts* de Python y utiliza internamente varias bibliotecas como scipy, numpy, scikitlearn, pysptk, sounddevice, os y programas como Praat [2]. A la hora de ser utilizada por nosotros ha tenido que ser configurada para su correcto funcionamiento en nuestro entorno. Los detalles de la configuración y los cambios realizados se dan en el manual del programador.

Os

DEFINIR.

Keras

DEFINIR.

VGGish

DEFINIR.

VGGish2Keras

DEFINIR.

4.5. Praat

Praat[3] [2] es un programa el cual nos permite realizar análisis fonéticos de audios vocales. Esta herramienta está enfocada a la investigación del habla. Permite hacer una multitud de análisis diferentes entre los que se encuentran análisis del discurso (análisis espectrales, análisis de intensidad, de formantes...) o análisis estadístico. Una característica importante para nosotros es que permite ser ejecutado mediante línea de comandos con diferentes parámetros. Otro aspecto interesante es que Praat tiene un *wrapper* para Python llamado Parselmouth [11], aunque nosotros no lo utilizamos en el proyecto. Ha sido desarrolada por Paul Boersma y David Weenink de la Universidad de Ámsterdam.

Será utilizada internamente por la biblioteca Disvoice para analizar una serie de características de los audios, que posteriormente volverá a procesar

[3] http://www.fon.hum.uva.nl/praat/

Disvoice con diferentes métodos de Python para devolvernos a nosotros las características finales deseadas.

4.6. Git

DEFINIR.

4.7. Github

DEFINIR.

4.8. ZenHub

DEFINIR.

4.9. TortoiseGit

DEFINIR.

4.10. Latex

DEFINIR.

MiKTeK

DEFINIR.

LaTeX

DEFINIR.

Aspectos relevantes del desarrollo del proyecto

5.1. Inicio del proyecto

Este proyecto se presentó como un proyecto de investigación sobre la extracción de biomarcadores de la voz para la detección de enfermedades neurodegenerativas o depresión. Al principio, se carecía de un objetivo concreto, debido a la incertidumbre inicial de qué camino se debía seguir y cómo iba a estar de avanzado este área de investigación.

Tras recopilar información, tanto de artículos científicos, como de otras fuentes, se llegó a la conclusión de que el objetivo del proyecto era mejor que estuviera relacionado con la enfermedad del Parkinson. Valoramos diferentes enfermedades como alzheimer, depresión y ELA. Elegimos la enfermedad del Parkinson debido a que la investigación de la detección de esta enfermedad a través de la voz estaba más avanzada y hay muchos artículos recientes y noticias de la realización actual de proyectos en este campo.

5.2. Investigación del proceso a seguir

Una vez establecida la enfermedad decidimos el proceso a seguir para conseguir un modelo correcto de clasificación de la enfermedad. El proceso a grandes rasgos estaba claro: extraer características de audios y utilizarlas para crear un clasificador. Pero antes, había que decidir varios puntos importantes en el proceso a seguir: ¿De qué audios se deben extraer las características? ¿De dónde íbamos a obtener esos audios? ¿Qué tipo de

pre-procesamiento requieren los audios? ¿Qué características se sacan de cada uno de ellos?...

Antes de todo, cabe destacar que como uno de los objetivos era hacer un estudio de investigación sobre diferentes algoritmos para la clasificación de los audios, necesitamos obtener un conjunto de audios de un proyecto concreto para poder comparar nuestros resultados de manera objetiva con ese proyecto concreto. Por ello, el proceso que íbamos a seguir podía estar influenciado de manera directa por el conjunto de datos que se nos prestara.

A la hora de intentar responder a las anteriores preguntas, nos documentamos a través de los artículos más importantes en este campo, con el objetivo de obtener las ideas más relevantes de cada uno de ellos, para decidir el enfoque de nuestro proceso. Los grupos de investigación más importantes se correspondían con dos grupos diferentes de investigadores, los cuales tienen artículos relevantes sobre este tema. La explicación en detalle se dará en el apartado *Trabajos Relacionados* 5.10, sin embargo aquí haremos eco de las ideas más importantes. Un grupo es el compuesto por Max A. Little y Tsanas Thanasis con artículos como [12]. Se puede obtener una serie de ideas principales de este grupo:

- Utilizan únicamente audios de **vocales sostenidas** para la obtención de características.
- Sostienen que un conjunto pequeño de características (<20) es suficiente para una correcta clasificación de los audios. Incluso [23] está relacionado íntimamente con esta idea, ya que a partir de un conjunto grande de características utiliza diferentes técnicas de selección de características y demuestra que con un conjunto menor que 20 se obtienen buenos resultados.

Otro grupo, también destacado, es el de J. R. Orozco-Arroyave J. D. Arias-Londoño y J. F. Vargas-Bonilla, con artículos como [15]. La idea más importante que podemos obtener es la siguiente:

- La pronunciación de **consonantes en discurso corrido** (frases, textos, palabras...) **aporta mucha información** de la pronunciación debido a la intervención de diferentes músculos necesarios para ésta. Por ello, se deben analizar otros tipos de audios para obtener más información que si analizamos únicamente pronunciación de vocales sostenidas.

- Cada tipo de audio debe ser utilizado para hacer un clasificador diferente. No se pueden utilizar diferentes tipos de audio dentro del mismo clasificador, ya que obtenemos resultados confusos, derivados de las diferentes pronunciaciones de diferentes palabras, frases, etc.

Condensando ambas, llegamos a la conclusión de que debíamos analizar diversa variedad de audios (ya que aportan más información) sin obsesionarnos por obtener un número inmenso de características de cada uno. Por ello obtendremos variedad de clasificadores que se corresponderán con la variedad de tipos de audio que tengamos y haremos un estudio sobre ellos. Los temas de qué audios utilizar, como pre-procesarles o que características obtener de cada uno se abordará en los siguientes apartados.

5.3. Conjunto de datos

El conjunto de datos de audios usado es el descrito en [14]. Como se explica en el artículo, es un conjunto de audios en castellano de 100 personas: 50 de ellas pacientes con Parkinson (PD) y 50 de ellas personas sanas (HC). Es un conjunto de datos realizado de la manera más balanceada posible, a parte de contener 50PD-50HC, también está balanceado en cuanto a sexo y edad tanto dentro de los 50 PD como dentro de los 50 HC. Todos los pacientes han sido diagnosticados por expertos y los audios vienen etiquetados con 3 diferentes medidas: PD/HC, UPDRS-III [7] y Hoehn & Yahr scale [10]. El conjunto de audios contiene un total de 4200 audios divididos en los siguientes tipos:

Monólogos 50 monólogos de PD y 50 de HC. El contenido es discurso libre sobre la respuesta a la pregunta *¿Qué haces cuando te levantas por la mañana?*. La duración de este tipo de audios comprende desde 00:30 a 02:30.

Texto leído 50 audios de PD y 50 de HC. El contenido es el siguiente texto balanceado: *Ayer fui al médico. ¿Qué le pasa? Me preguntó. Yo le dije: Ay doctor! Donde pongo el dedo me duele. ¿Tiene la u~ña rota? Sí. Pues ya sabemos qué es. Deje su cheque a la salida.*

Vocales 750 audios de PD y 750 de HC. Pronunciación sostenida de cada una de las 5 vocales 3 veces por persona. 150 audios de HC y 150 de PD por cada vocal.

Palabras 1250 audios de PD y 1250 de HC. Pronunciación de palabras para el análisis silábico. Cada persona tanto PD como HC pronunciará 25 palabras diferentes: *brasa, coco, petaca, etc.*

5.4. Resumen general del estudio

En esta sección se hará una mera introducción inicial al estudio realizado.

Un aspecto importante del proyecto es el siguiente. En un primer momento se planeó la extracción de características, creación de los clasificadores y finalmente utilización de ellos. Para ello comenzamos, en la **primera fase** de experimentos, extrayendo las características (que comentaremos posteriormente) con la biblioteca **Disvoice**. Creamos los diferentes clasificadores para esas características y obtuvimos resultados bastante inferiores a los recogidos en el estado del arte, i.e. [15]. Destacamos que en las diferentes fases del estudio, lo que hemos ido variando han sido los conjuntos de datos extraídos de cada audio, realizando los mismos experimentos con clasificadores (o modificando los experimentos muy poco).

Como no obtuvimos los resultados esperados, decidimos hacer una mejora a los experimentos, una **segunda fase** del estudio. Esta mejora consistía en dos partes. La primera de ellas fue añadir para cada instancia los atributos **edad y sexo** del paciente a las características extraídas por Disvoice. La segunda mejora fue separar las instancias del conjunto de datos entre hombres y mujeres. Se planteó de esta manera, debido a que [15] hace validación cruzada estratificada según dos elementos: Edad y clase (PD: *Parkinson Disease* o HC: *Healthy control*). Con la biblioteca utilizada para los experimentos, *scikitlearn*, solamente se puede estratificar según 1 atributo. Separando los conjuntos de datos por sexos y estratificando cada sexo por clase, estamos simulando esa validación cruzada estratificada por 2 atributos que utiliza ese artículo. Se mejoraron los resultados obtenidos con los conjuntos de datos de la primera fase, pero aun así estaban lejos de los resultados en el estado del arte.

Con intención de mejorar los resultados y dar una perspectiva diferente a los experimentos, realizamos una **tercera fase** de los mismos. En esta tercera fase utilizamos la biblioteca de *Deep Learning* llamada **VGGish** CITAR, REF. Consiste en la extracción de características mediante la una red neuronal pre-entrenada con audios de *Youtube*, que utiliza capas convolucionales. Al estar pre-entrenada, ya tenemos los pesos para la extracción de características y, por tanto, lo único que debemos realizar es utilizar las funciones de extracción de esa biblioteca. En esta fase, sacamos para

cada audio otros dos conjuntos de características, con los cuales realizamos los experimentos con los clasificadores. Los resultados obtenidos seguían estando en la magnitud de los obtenidos por nosotros, pero sin llegar a los mejores del estado del arte.

Como no mejoraban, decidimos hacer la fase de desarrollo de una web-app que hiciera uso del mejor clasificador... COMENTAR POR ENCIMA LO HECHO DEL SP8 (incluido) EN ADELANTE.

5.5. Metodología del estudio

5.6. Primera Fase: atributos Disvoice

Nuestro estudio comprenderá la comparación de diferentes clasificadores construidos cada uno con diferentes conjuntos características para cada tipo de audio. Utilizaremos 3 diferentes tipos de audio: **vocales sostenidas, 5 palabras diferentes y texto leído**. Qué características sacamos para cada tipo de audio se presentará en la subsección *Modelado del discurso* 5.6. El proceso para la realización de los clasificadores es el siguiente (ver figura 5.3):

1. Pre-procesamiento de los audios: preparación de audios para la extracción de diferentes medidas.
2. Modelado del discurso: extracción de diferentes tipos de características para cada tipo de audios.
3. Clasificación: construir diferentes clasificadores para cada conjunto de características para hacer un estudio comparativo.

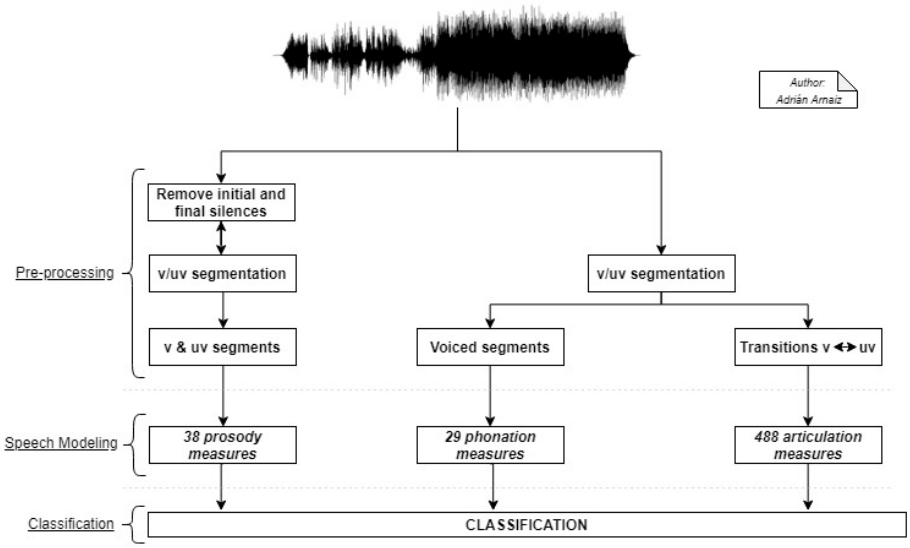

Figura 5.3: Esquema del proceso para abordar los experimentos.

Pre-procesado de audios

En esta etapa se realizan 2 tareas principales: la eliminación de sonidos inicial y final de los audios y la segmentación en fragmentos con voz y sin voz de los mismos (los llamaremos segmentos *voiced* y *unvoiced*). La eliminación de sonidos inicial y final de los audios se realiza ya que a la hora de extraer las medidas de prosodia se puede tener algunos incovenientes si los silencios iniciales son muy largos. Si fueran excesivamente largos tendríamos resultados erróneos en las características, como, por ejemplo, las relacionadas con la duración promedio de silencios, la variabilidad de la duración de los silencios y otras medidas que se calculan sobre las pausas. Para la extracción de medidas de fonación y articulación este proceso lo realizan internamente los *scripts* de la biblioteca Disvoice [16] utilizando Praat. Sin embargo, a la hora de obtener las medidas prosódicas es necesario realizar la eliminación de manera previa.

La **segmentación en fragmentos *voiced* y *unvoiced*** se realiza para analizar el discurso, es decir, se sacarán medidas que necesitan de esta fragmentación (i.e. *Jitter* de los fragmentos *voiced* o MFCC de las transiciones entre *voiced* y *unvoiced*). Esta tarea la hacen internamente los *scripts* de la biblioteca Disvoice utilizando Praat.

Caract.	Número	Breve descripción
1ª derivada F0	1x4=4	1ª deriv. frec fundamental
2ª derivada F0	1x4=4	2ª deriv. frec fundamental
Jitter	1x4=4	Perturbación de F0
Shimmer	1x4=4	Perturbación de Amplitud
APQ	1x4=4	Cociente de perturb. de amplitud
PPQ	1x4=4	Cociente de perturb. de periodo
Energía Log	1x4=4	Explicado en [1]
Grado unvoiced	1	Grado *unvoiced*

Tabla 5.1: Características de fonación. En detalle en [16].

Modelado del discurso

Se extraerán 3 diferentes conjuntos de características para cada tipo de audio: de fonación, de articulación y prosódicas. Como hemos visto en nuestro conjunto de datos, tenemos texto leído, palabras, vocales y monólogos. En este proyecto utilizaremos el texto leído, las 5 palabras con mejor resultado en [15] (*atleta, campana, braso, gato, petaca*) y las 5 vocales. El monólogo no será utilizado debido a que al ser discurso libre y no predefinido, las características extraídas dependen también de cómo sea el discurso (i.e. las palabras que se digan, las pausas...).

Medidas de fonación

De las medidas de fonación obtendremos 11 conjuntos diferentes: 1 para el texto leído, 5 para las palabras (1 por palabra elegida) y 5 por vocal (1 por vocal). En total, de cada audio se sacan un conjunto de **29 medidas** basadas en la perturbación de la fonación. Las medidas de fonación, son extraídas de los segmentos *voiced*, utilizando para ello la biblioteca Disvoice (`phonation.py`). Estas características son descritas en la tabla 5.1.

Obtenemos un vector de 29 características: las 7 medidas por sus 4 funcionales(media m, desviación std, curtosis k y oblicuidad sk) + grado de *unvoiced* [a].

[a] El grado de unvoiced es el ratio entre la duración de los segmentos sin voz entre la duración total del audio [16].

Caract.	Número	Breve descripción
BBE onset	22x4=88	22 coef. BBE de trans. *v ->uv*
MFCC onset	12x4=48	12 coef. MFCC de trans. *v ->uv*
1ªD MFCC onset	12x4=48	1ª deriv. 12 coef. MFCC de trans. *v ->uv*
2ªD MFCC onset	12x4=48	2ª deriv. 12 coef. MFCC de trans. *v ->uv*
BBE offset	22x4=88	22 coef. BBE de trans. *uv ->v*
MFCC offset	12x4=48	12 coef. MFCC de trans. *uv ->v*
1ªD MFCC offset	12x4=48	1ª deriv. coef. 12 MFCC de trans. *uv ->v*
2ªD MFCC offset	12x4=48	2ª deriv. coef. 12 MFCC de trans. *uv ->v*
1ª formante F0	1x4=4	1ª formante de frecuencia
1ªD 1ª formante F	1x4=4	1ª deriv. 1ª formante de frecuencia
2ªD 1ª formante F	1x4=4	2ª deriv. 1ª formante de frecuencia
2ª formante F	1x4=4	2ª formante de la frecuencia
1ªD 2ª formante F	1x4=4	1ª deriv. 2ª formante de frecuencia
2ªD 2ª formante F	1x4=4	2ª deriv. 2ª formante de frecuencia

Tabla 5.2: Características de articulación. En detalle en [16].

Medidas de articulación

De las medidas de articulación obtendremos 6 conjuntos diferentes: 1 para el texto leído y 5 para las palabras (1 por palabra elegida). En total de cada audio se sacan un conjunto de **488 medidas** de articulación. Las medidas de articulación son extraídas de las transiciones entre los segmentos *voiced* y *unvoiced* utilizando para ello la biblioteca Disvoice (`articulación.py`). Estas características son descritas en la tabla 5.2

> Obtenemos un vector de 488 características: las 122 medidas por sus 4 funcionales(media *m*, desviación *std*, curtosis *k* y oblicuidad *sk*).

Medidas de prosodia

De las medidas de prosodia obtendremos 1 conjuntos para el texto leído. En total de cada audio se sacan un conjunto de **38 medidas** basadas en la duración, la frecuencia fundamental, la energía y ratios de la composición del audio en lo relativo a segmentos *voiced* y *unvoiced*. Las medidas de prosodia son extraídas del audio completo, tanto segmentos *voiced* como *unvoiced*, utilizando para ello la librería Disvoice (`prosodia.py`). Estas características son descritas en la tabla 5.3.

Caract.	Número	Breve descripción
Frec. fundamental	7	relativas a la frec. fundamental
Energía	9	9 medidas relativas a la energía
Ratios *v-uv*	22	22 medidas relativas a *v-uv*

Tabla 5.3: Características de prosodia. En detalle en [16].

Obtenemos un vector de 38 características, las descritas en la tabla 5.3. Esta vez sin sacar los funcionales para cada medida.

Conjuntos de datos totales

Nos salen 18 total. En total 18 (subsets de características). 3xRT+2x. describir los conjuntos de datos obtenidos: ART_RT, PHON_W_GATO.

Experimentos clasificadores

Se explicará cuando se realice. Resumen de lo que se dirá: Elegir un conjunto de algoritmos de clasificación. Realizar un clasificador de cada tipo con cada conjunto de ccas extraídas (18 total). En total 18 (subsets de características) X N (clasificadores elegidos). Comentar que algoritmos de clasificación hemos elegido, con qué parámetros. Comentar como se realiza la cross-validation y cómo serán evaluados (accuracy, Roc...). CÓMO IMPLEMENTARLO

Resultados

Resultados de la primera fase del proyecto

5.7. Segunda Fase: Disvoice modificado

Comentar POR QUÉ, CÓMO IMPLEMENTARLO, +edad y sexo, mujer, hombre, MIRAR RESUMEN GENERAL

Modelado del discurso

Comentar que es igual que antes pero añadiendo dos y dividiendo para obtener los datos. MIRAR RESUMEN GENERAL. CÓMO IMPLEMENTARLO

Conjuntos de datos totales

describir los conjuntos de datos obtenidos: ART_RT_MUJER+E/HOMBRE+E/Edad+Se PHON_W_GATO...

Experimentos clasificadores

Explicación de lo realizado. LOS DE ANTES MÁS QUÉ AÑADIMOS. CÓMO IMPLEMENTARLO

Resultados

Resultados de la primera fase del proyecto. TABLAS, GRAFICOS.

5.8. Tercera Fase: VGGish

MIRAR RESUMEN GENERAL, POR QUÉ, CÓMO, embeddings y espectros, FALLO <0.975

Modelado del discurso

Comentar que es igual que antes pero añadiendo dos y dividiendo para obtener los datos. MIRAR RESUMEN GENERAL. CÓMO IMPLEMENTARLO

Conjuntos de datos totales

describir los conjuntos de datos obtenidos: embeddings, espectros: nombre de cada subconjunto

Experimentos clasificadores

Explicación de lo realizado. LOS DE ANTES MÁS QUÉ AÑADIMOS. CÓMO IMPLEMENTARLO

Resultados

Resultados de la primera fase del proyecto. TABLAS, GRAFICOS.

5.9. Estudio comparativo entre clasificadores - Cual elegir

Se explicará cuando se realice. Resumen de lo que se dirá: Comentar el resultado concreto de los clasificadores explicados en el apartado anterior. Realizar comparativa entre los clasificadores anteriores. Mostrar tablas de resultados, conclusiones..., TABLAS, GRAFICOS.

5.10. Siguientes pasos, web, app, docker...

Contar que se ha hecho a partir del SP8 (incluido)

Trabajos relacionados

En este apartado se comentará el estado del arte de la materia y algunos trabajos relacionados.

Para comenzar, comentaremos que hay dos grupos de investigación importantes en este campo de estudio. El primer grupo sería el encabezado por el autor M. A. Litlle de la Universidad de Aston, Birmingham, Inglaterra. Este autor a su vez encabeza un proyecto para la obtención de una gran base de datos de 10000 audios obtenidos de pacientes con Parkinson. Esta iniciativa es conocida como Parkinson Voice Inititative[4]. Este conjunto de autores tiene varios artículos [22], [23] y el más citado [12]. Estos artículos aportan un buen análisis sobre las características a extraer de los audios (lineares y no lineares) y algoritmos de selección de características. Los aspectos más relevantes de cada artículo son los siguientes:

- *Suitability of dysphonia measurements for telemonitoring of Parkinson's disease* [12].
  Este artículo trata sobre la aplicación de diferentes medidas estándares (lineales) y no estándares (no lineales) de disfonía para la clasificación automática de personas con Parkinson y personas sanas. Además, añaden otra nueva medida, calculada en este mismo artículo, llamada PPE, *pitch period entropy*. Se centra en responder a la pregunta de qué características son las mejores para la detección del Parkinson. Para ello utiliza únicamente audios de vocales sostenidas (28 PD y 8 HC). Se calculan un total de 17 medidas de disfonía de cada audio usando diferentes software como Praat [2]. Posteriormente, se hace un

[4] http://www.parkinsonsvoice.org/

análisis de correlación entre las medidas obtenidas, y de aquellos pares que tienen un coeficiente de correlación mayor del 95 %, se elimina una. Después de este análisis se obtienen 10 características: *jitter* medido diferencia absoluta, *jitter* medido como media entre ciclos, APQ, *shimmer* (calculado como la diferencia absoluta promedio entre las amplitudes de los períodos consecutivos), HNR, NHR, RPDE, DFA, la dimensión de correlación y el comentado PPE.

En el siguiente paso del proceso, utilizan el algoritmo de aprendizaje supervisado SVM con kernel de base radial, para construir el modelo. Esto se realiza con cada uno de los 1023 diferentes subconjuntos posibles de las 10 características, $\sum_{i=1}^{10} \binom{10}{i}$, para encontrar el mejor subconjunto posible. Esto es debido a que los autores consideran que es un número pequeño de conjuntos y se pude hacer búsqueda exhaustiva. Se llega a la conclusión que el **mejor conjunto de características es el formado por HNR, RPDE, DFA, y PPE** que devuelve una precisión del **91 %**, seguido en precisión por el conjunto completo de las 10 características.

- *Accurate Telemonitoring of Parkinsons Disease Progression by Noninvasive Speech Tests* [22].
  En este artículo trata sobre cómo **el objetivo de la clasificación del Parkinson puede ser de regresión**. En vez de clasificar entre personas sanas (HC) y con Parkinson (PD), lo que se hace es medir el nivel de parkinson de una persona usando la escala UPDRS (*Unified Parkinson's Disease Rating Scale*). Para ello saca un total de 16 características de audios con vocales sostenidas, de las que se hace un análisis de correlación pero no se elimina ninguna.

  Para la construcción de modelos se utilizan 4 técnicas diferentes: 3 de ellas de regresión lineal (LS, IRLS y LASSO) y una de regresión no lineal (CART's). Se llega a la conclusión de que los métodos lineales no dan malos resultados, siendo el IRLS el mejor de los 3. Sin embargo, el que mejor precisión da es el método CART. Los errores son de 8.47 ± 0.17 para UPDRS total con IRLS y de 7.52 ± 0.25 usando el método CART. A parte de estos resultados, en el artículo se realiza un análisis de la correlación de las características fijándose en los coeficientes devueltos por el método LS. En ellos se puede ver como las características altamente correlacionadas tienen magnitud similar y signo opuesto.

- *Novel Speech Signal Processing Algorithms for High-Accuracy Classification of Parkinson's Disease* [23].

En este artículo el objetivo es clasificar entre personas sanas y con PD a partir de audios de vocales sostenidas **utilizando un gran conjunto de medidas de disfonía extraídas de los audios**. Un punto novedoso de este experimento, es que hasta ese momento, se habían elegido conjuntos de pocas (<20) medidas de disfonía y medido su correlación. En este artículo, trata un total de 132 medidas lineares y no lineares de cada audio a las que se aplicará diferentes algoritmos de selección de características (algoritmos FS) para elegir varios conjuntos (uno por cada algoritmo). Los algoritmos de selección de características han sido: LASSO, RELIEF, mRMR y LLBFS. Cada algoritmo de selección de características ha elegido un conjunto diferente, siendo básicamente medidas como *jitter* y *shimmer*, variantes de medidas de ruido, MFCCs y medidas no lineales. Se ha analizado el número de características que comprende cada conjunto y se ha llegado a la conclusión de que cada conjunto tendrá 10 medidas únicamente. Según se explica, esto es debido a que usando más de 22 características se tiende al sobreajuste y que tampoco hay mucha mejoría usando 22 características en lugar de 10.

Posteriormente cada uno de esos conjunto de características ha sido usado para construir dos clasificadores con dos métodos distintos: random forest y SVM con kernel Gaussiano.Se obtienen resultados que mejoran cualquier artículo de de accurracy del 97.7 % utilizando las 132 características con SVM. Sin embargo y como acabamos de explicar anteriormente, en el mismo artículo acusa del accuracy tan alto a un posible sobreajuste al usar tantas caractarísticas. Por lo explicado anteriormente, cada algoritmo FS escoge un conjunto de 10 medidas elegidas. El mejor resultado se da para el subconjunto de características escogido por el algoritmo RELIEF y usando SVM, presentando una accuracy del 98.6 % frente al RF que con el mismo conjunto llega a un accuracy del 93.5 %.

Para terminar con el resumen de este artículo, indicar que la importancia está en que se trabaja con un número grande de medidas (132), llegando a la conclusión de que los subconjuntos idóneos tendrán alrededor de 10 características ya que tienen resultados parecidos a los conjuntos grandes pero generalizarán mejor que éstos.

El segundo grupo de investigación y para nosotros más importante, ya que seguimos los pasos para la extracción de características, es el liderado por J. R. Orozco-Arroyave de la Universidad de Antioquía, Colombia. Estos artículos son más recientes que los de el anterior grupo de investigación y,

aunque tienen menos repercusión, son autores que trabajan y colaboran con el MIT (Massachusetts Institute of Technology) el cual tiene gran prestigio. El artículo más importante del grupo y que aporta una nueva visión a esta materia es [15]. Esta nueva visión es debido a que toma en cuenta más audios aparte de únicamente de la vocal sostenida. También tiene en cuenta varios idiomas. Es precisamente de este artículo del que hemos querido seguir los pasos a la hora de determinar qué características sacar de los audios y cómo sacarlas. También valorar y agradecer a este grupo de investigación la cesión de los audios descritos en [14], un corpus de audios de vocales sostenidas, monólogos, frases y palabras recopilados de un total de 100 personas (50 sanas y 50 PD) que contiene aproximadaamente 4200 audios. Analizamos el artículo más importante de este grupo:

- *Automatic detection of Parkinson's disease in running speech spoken in three different languages* [15].
  En este artículo clasifica entre personas sanas y con PD y se incorporan a la materia varias novedades relacionadas con los tipos de audios a utilizar.La primera de ellas es utilizar audios de monólogos, texto leído y palabras aparte únicamente de la vocal sostenida. Esto es debido a que la mayoría de los experimentos se basan en vocales sostenidas, obviando los consejos de los neurólogos respecto a que las consonantes requieren un mejor control de los movimientos de órganos (la lengua por ejemplo) y de ahí se puede sacar mucha más información. Por ello se utiliza un método para la caracterización de las señales de voz, basado en la segmentación automática de las expresiones en cuadros con voz y sin voz.

  La otra gran novedad que incorpora es la utilización de corpus de audios de varios idiomas a la vez, en este caso 3 diferentes. Esto se utiliza, por ejemplo, para lo que ellos mismos llaman *cross-language experiments*. En estos experimentos, el entrenamiento del modelo se hace en un idioma y el testeo con otro idioma diferente. En este artículo se realizan diferentes experimentos, construyendo varios modelos para cada tipo de audio y así evaluar qué tipo de audio es el mejor para la clasificación. Se comienza haciendo un preprocesado de los audios donde los silencios iniciales y finales son eliminados. Se prosigue dividiendo los audios en partes con voz y partes sin voz (estos frames con voz y sin voz se corresponden a los frames donde el software Praat [2] detecta que hay discurso y donde no lo hace). Para acabar el preprocesado se eliminan los fragmentos menores a 40 ms.

  A continuación empiezan con la extracción de características de los

audios. Se sacan 3 grupos de medidas diferentes con las que se construyen 3 modelos diferentes: 2 sacando diferentes tipos de medidas de los fragmentos con voz y 1 sacando medidas de los fragmentos sin voz. De los fragmentos con voz, se sacan por un lado 64 medidas prosódicas (*jitter*, *shimmer*...) con sus estadísticos (media, desviación, coeficiente de asimetría y curtosis). Por otro se sacan NHR, NNE, GNE, *F1*, *F2* y las 12 MFCC con sus estadísticos. De los frames sin voz se hace un análisis cepstral y de energía para calcular las 12 MFCC y 25 BBE con sus estadísticos.

Después se hace la clasificación utilizando SVM con kernel Gaussiano, que utiliza *grid search* para los parámetros C que mide la estrictez del margen y γ que es un parámetro del kernel Gaussiano utilizando como medida objetivo *accuracy*. Se utiliza una técnica de validación cruzada 10-fold.

Como resultado obtenemos 3 modelos diferentes para cada tipo de audio. De cada modelo saca 4 medidas para poderles comparar: *accuracy*, *sensitivity*, *specificity* y *AUC* (área bajo la curva de ROC). En el artículo compara los experimentos por tipo de audio. El mejor rendimiento lo consiguen, el texto leído, ya que consigue, para el idioma castellano y con el modelo construido con las medidas extraídas de los frames sin voz, un accuracy del 97 % y un AUC del 99 %.

Ahora trataremos dos trabajos de una base da datos adquirida. Estos trabajos corresponden a Giovanni Dimauro de la Universidad de Bari, Italia. En el segundo de estos trabajos se aborda una perspectiva diferente para la diagnosis de Parkinson, utilizando sistemas Speech-To-Text:

- *VoxTester, software for digital evaluation of speech changes in Parkinson disease* [5].
  Este artículo describe la realización de una herramienta software para la ayuda en la evaluación de los cambios y variaciones en la voz de pacientes con la enfermedad del Parkinson. El funcionamiento de la herramienta es el siguiente: insertamos una serie de audios del paciente, la herramienta analiza el audio y nos devuelve 4 gráficas que un experto deberá interpretar para la evaluación de la enfermedad. Las gráficas de salida son las siguientes: gráfica del DDK *rate*, gráfica de la intensidad vocal y duración del discurso, gráfica del espectro de frecuencias vocal y gráfica del nivel de presión vocal. Estos parámetros sacados de la voz son importantes ya que pueden ser indicadores de la enfermedad del Parkinson. Por ejemplo, un rango de frecuencias de

Artículos	Clasificación PD/HC	Regresión Mapeo UPDRS	Otro análisis
[12] [23] [15]	X		
[22]		X	
[5] [4]			X

Tabla 6.4: Objetivo de cada artículo

la voz pequeño, una frecuencia fundamental baja o un decaimiento notable de la intensidad de la voz en el discurso serán una de las características del discurso para personas con esta enfermedad.

En este experimento no se utiliza en ningún momento técnicas de clasificación ni tampoco la herramienta discierne por ella misma si la persona del audio tiene una enfermedad o en qué grado. Lo único que hace es analizar la onda sonora y mostrar algunas características, por lo que no se alinea del todo con nuestros objetivos.

- *Assessment of Speech Intelligibility in Parkinson's Disease Using a Speech-To-Text System* [4].
  Este experimento trata de resolver el problema del anterior (valuación de los cambios y variaciones en la voz de pacientes con la enfermedad del Parkinson) con otra línea. Para ello utilizará sistemas de reconocimiento de voz y transcripción a texto (*Speech-To-Text Systems*) para evaluar la inteligibilidad del discurso de las personas con Parkinson. El proceso será el siguiente: la persona con Parkinson habla a un sistema STT, el texto de la transcripción se pasa a un programa que evalúa su acierto y finalmente ese programa devuelve el porcentaje de fallo en el reconocimiento de voz respecto a la frase objetivo. El objetivo de este proyecto ha sido comparar los fallos de reconocimiento en las palabras entre 3 grupos de personas: jóvenes sanos, mayores sanos y pacientes de la enfermedad del Parkinson. La conclusión a la que llegan los investigadores es que había mucho mas fallo en el reconocimiento de palabras en los pacientes de la enfermedad del Parkinson.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

[1] *Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms*, volume ETSI ES 201 108 V1.1.3. European Telecommunications Standards Institute, September 2003.

[2] Paul Boersma and David Weenink. Praat, a system for doing phonetics by computer. *Glot international*, 5:341–345, 01 2001.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Giovanni Dimauro, Vincenzo Di Nicola, Vitoantonio Bevilacqua, Danilo Caivano, and Francesco Girardi. Assessment of speech intelligibility in parkinson's disease using a speech-to-text system. *IEEE Access*, pages 22199–22208, 2017.

[5] Giovanni Dimauro, Vincenzo Di Nicola, Vitoantonio Bevilacqua, Francesco Girardi, and Vito Napoletano. Voxtester, software for digital evaluation of speech changes in parkinson disease. *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, pages 1–6, 2016.

[6] Theodoros Evgeniou and Massimiliano Pontil. Support vector machines: Theory and applications. volume 2049, pages 249–257, 01 2001.

[7] SRLE Fahn. Unified parkinson's disease rating scale. *Recent development in Parkinson's disease*, 1987.

[8] Salvador García, Sergio Ramírez-Gallego, Julián Luengo, José Manuel Benítez, and Francisco Herrera. Big data preprocessing: methods and prospects. *Big Data Analytics*, 1, 2016.

[9] Aileen K Ho, Robert Iansek, Caterina Marigliani, John L Bradshaw, and Sandra Gates. Speech impairment in a large sample of patients with parkinson's disease. *Behavioural neurology*, 11(3):131–137, 1999.

[10] Margaret M Hoehn and Melvin D Yahr. Parkinsonism: onset, progression, and mortality. *Neurology*, 17(5):427–427, 1967.

[11] Yannick Jadoul, Bill Thompson, and Bart de Boer. Introducing Parselmouth: A Python interface to Praat. *Journal of Phonetics*, 71:1–15, 2018.

[12] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman, and L. O. Ramig. Suitability of dysphonia measurements for telemonitoring of parkinson's disease. *IEEE Transactions on Biomedical Engineering*, 56:1015–1022, Enero 2009.

[13] Janine Möbes, Gregor Joppich, Frank Stiebritz, Reinhard Dengler, and Christine Schröder. Emotional speech in parkinson's disease. *Movement Disorders*, 23(6):824–829, 2008.

[14] J. R. Orozco-Arroyave, J. D. Arias-Londoño, J. F. Vargas-Bonilla, M. González-Rátiva, and E. Nöth. New spanish speech corpus database for the analysis of people suffering from parkinson's disease. *Proceedings of the 9th Language Resources and Evaluation Conference (LREC)*, pages 342–347, 2014.

[15] J. R. Orozco-Arroyave, F. Hönig, J. D. Arias-Londoño, J. F. Vargas-Bonilla, K. Daqrouq, S. Skodda, J. Rusz, and E. Nöth. Automatic detection of parkinson's disease in running speech spoken in three different languages. *The Journal of the Acoustical Society of America*, 139:481–500, Enero 2016.

[16] Juan Rafael Orozco-Arroyave, Juan Camilo Vásquez-Correa, Jesús Francisco Vargas-Bonilla, Raman Arora, Najim Dehak, Phani S Nidadavolu, Heidi Christensen, Frank Rudzicz, Maria Yancheva, H Chinaei, et al. Neurospeech: An open-source software for parkinson's speech analysis. *Digital Signal Processing*, 77:207–221, 2018.

[17] Kathe S Perez, Lorraine Olson Ramig, Marshall E Smith, and Christopher Dromey. The parkinson larynx: tremor and videostroboscopic findings. *Journal of Voice*, 10(4):354–361, 1996.

[18] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. *Encyclopedia of Database Systems*, 532–538:532–538, 01 2009.

[19] J Rusz, R Cmejla, H Ruzickova, and E Ruzicka. Quantitative acoustic measurements for characterization of speech and voice disorders in early untreated parkinson's disease. *The journal of the Acoustical Society of America*, 129(1):350–367, 2011.

[20] Sabine Skodda, Wenke Visser, and Uwe Schlegel. Vowel articulation in parkinson's disease. *Journal of voice*, 25(4):467–472, 2011.

[21] Glenn T Stebbins and Christopher G Goetz. Factor structure of the unified parkinson's disease rating scale: motor examination section. *Movement disorders: official journal of the Movement Disorder Society*, 13(4):633–636, 1998.

[22] A. Tsanas, M. A. Little, P. E. Mcsharry, and L. O. Ramig. Accurate telemonitoring of parkinsons disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering*, 57:884–893, Abril 2010.

[23] A. Tsanas, M. A. Little, P. E. Mcsharry, J. Spielman, and L. O. Ramig. Novel speech signal processing algorithms for high-accuracy classification of parkinson's disease. *IEEE Transactions on Biomedical Engineering*, 59:1264–1271, Mayo 2012.

[24] Wikipedia. Jitter — wikipedia, la enciclopedia libre, 2019. [Internet; descargado 16-abril-2019].

[25] Wikipedia. Mfcc — wikipedia, la enciclopedia libre, 2019. [Internet; descargado 16-abril-2019].

[26] I. H. Witten, E. Frank, M. A. Hall, , and C. J Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 4 edition, 2017.

[27] Eberhard Zwicker and Ernst Terhardt. Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *The Journal of the Acoustical Society of America*, 68(5):1523–1525, 1980.