UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE MÁSTER

DESARROLLO DE UN ALGORITMO CON MACHINE LEARNING PARA LA CLASIFICACIÓN DE PACIENTES CON PARKINSON.

Máster Universitario en Bioinformática y biología computacional

Autor: MARÍN_MÉNDEZ, Juan Jesús

Tutor: Sayar_Beristain, Onintza
Pharmamodelling SL
Ponente: MARTINEZ_MUÑOZ, Gonzalo
Escuela Politécnica Superior. Universidad Autónoma de Madrid

Madrid, Septiembre de 2019

Agradecimientos

Me gustaría agradecer a mis tutores Onintza y Gonzalo,
sin ellos este trabajo no podría haberse realizado.
A mis compañeros.

Índice

Resumen

El *Machine learning* (ML) es una herramienta de uso creciente en el área de la salud y clínica. El empleo de algoritmos matemáticos enfocados a la predicción nos permite obtener conocimiento a partir de datos y realizar predicciones a partir de nuevas instancias que se le suministren al algoritmo. Tras el Alzheimer, el Parkinson es la segunda enfermedad neurodegenerativa más prevalente. Existen síntomas no motores que afectan al paciente de una manera temprana como por ejemplo el deterioro en el habla. Existen técnicas no invasivas que valoran este deterioro en la voz y cuyos datos pueden emplearse para alimentar modelos de ML.

Con todo ello, se planteó el **objetivo** de desarrollar un modelo de ML a partir de datos de grabaciones de voz almacenados en repositorios públicos que permitan distinguir entre individuos con Parkinson y sanos.

Resultados: Se obtuvo un modelo de ML empleando la técnica de *Random forest* con una sensibilidad de 0.891 y una especificidad de 0.873. Este modelo obtuvo un *accuracy* de 0.866 en la validación cruzada y de 0.907 en la validación con un conjunto de test.

Conclusión: Se puede hacer uso de herramientas de ML para clasificar enfermedades complejas del tipo Parkinson empleando técnicas no invasivas.

Objetivos

El objetivo principal del proyecto es el de obtener un clasificador que permita distinguir entre sujetos que presentan la enfermedad de Parkinson y sujetos sanos. Para ello, se hará uso de *datasets* que contienen parámetros vocales recogidos de sujetos en diversos estudios y que serán analizados mediante técnicas de minería de datos y ML.

Para cumplir con el objetivo principal del proyecto, se realizarán las siguientes tareas:

- Obtención y pre-procesamiento de los conjuntos de datos (*datasets*). Se hará uso de bases de datos públicas localizadas en repositorios. Se contactará también con grupos de investigación que tengan datos no publicados y en caso de ser posible se hará uso de datos no públicos.
- En el caso de que finalmente se haga uso de diferentes bases de datos se estudiará la viabilidad de integración aplicándose técnicas para la integración de las diversas fuentes.
- Desarrollo de los algoritmos de ML para la clasificación.
- Evaluación de las métricas de clasificación de cada uno de los algoritmos obtenidos en el punto anterior para finalmente seleccionar el mejor algoritmo de clasificación.
- En caso de disponer de una muestra externa, validar el algoritmo.

Introducción

Como indica la Sociedad Española de Neurología, el Parkinson es la segunda patología neurodegenerativa más frecuente tras el Alzheimer. La lista de diagnósticos neurológicos más frecuentes en mayores de 65 años, también se encuentra liderada por el Alzheimer seguida del Parkinson. En nuestro país, el 70% de las personas diagnosticadas de Parkinson tienen más de 65 años. Este dato junto con el hecho del creciente envejecimiento de la población española, hace de la prevención de esta enfermedad neurológica un punto caliente para la investigación.

El Parkinson (PD, por sus siglas en inglés) es un trastorno degenerativo que afecta al sistema nervioso central. Es un trastorno bastante común afectando a 1-2 personas por cada 100 habitantes, y cuya prevalencia aumenta con la edad afectando al 1% de la población mayor de 60 años (1). El diagnóstico final del PD se realiza post-mortem durante una autopsia. En algunos de los casos, el diagnóstico post-mortem no confirma el diagnóstico clínico previo. Algunos estudios previos sugieren que el valor predictivo positivo era del 76%, pero con el tiempo ha ido mejorando. El incluir especialistas de los trastornos del movimiento ha producido un crecimiento del valor predictivo positivo llegando a ser del 98% (2). Estas variaciones en el valor predictivo, junto con las limitaciones en el acceso a especialistas, sugiere que deban revisarse los criterios diagnósticos o que se generen nuevas pruebas que ayuden a mejorar la exactitud clínica en el diagnóstico del PD.

Fuera del examen patológico del cerebro, no existen pruebas de laboratorio o técnicas de imagen que puedan ser empleadas en el diagnóstico de PD con certeza. Por ello, el diagnóstico es puramente clínico y basado en evaluaciones neurológicas combinadas con la historia médica previa. En resumen, la aparición de test adicionales basados en otros criterios, pueden ser de gran utilidad a la hora de evaluar el parkinsonismo (2).

Estudios de seguimiento de la evolución del PD con esta herramienta, sugieren que el curso del trastorno no es lineal y que el grado de deterioro es variable y más rápido en las fases tempranas del mismo (3).

La sintomatología principal del PD son los temblores, la rigidez, y otros trastornos del movimiento general. Otros síntomas con una gran prevalencia en el trastorno son el deterioro vocal, con una prevalencia descrita en diversos estudios de entre un 70-90% tras la aparición de la enfermedad (4). La prevalencia elevada, junto a ser uno de los indicadores más tempranos del trastorno, hace del estudio de los parámetros vocales, un campo de investigación prometedor en el diagnóstico temprano del Parkinson.

La telemonitorización es una herramienta que permite un screening coste efectivo de aquellas enfermedades donde la voz se ve afectada. Estas herramientas, pueden reducir las frecuentes visitas a la clínica y aliviar al sistema de salud (reduciendo por ejemplo las visitas presenciales) descendiendo costes e incrementando la precisión de la evaluación clínica (5).

El potencial de la telemonitorización recae en el diseño de un test simple que pueden ser auto-administrados de una manera rápida y remota. Las grabaciones de discursos son no invasivas y pueden ser integradas realmente en aplicaciones de la telemedicina, lo que les hace buenos candidatos (5).

¿Por qué el Machine learning?

En la segunda parte del siglo 20, el ML evolucionó como un subcampo de la inteligencia artificial (IA) que implicaba el desarrollo de algoritmos de *self-learning* para obtener conocimiento a partir de los datos con el objetivo de realizar predicciones. El ML ofrece una alternativa efectiva a la manera clásica (construir manualmente modelos de los que derivan reglas de diversos tipos) de obtener información de los datos. El ML, extrae conocimiento a partir de los datos para gradualmente mejorar el funcionamiento de los modelos predictivos y tomar decisiones dirigidas hacia un objetivo. Hoy en día, se está empleando el ML en el mundo sanitario dirigiéndonos rápidamente hacia una mejora del conocimiento de las enfermedades a partir de grandes cantidades de datos que han sido almacenando a lo largo del tiempo.

Según a qué tipo de problema nos enfrentamos, existen distintos tipos de ML: el aprendizaje supervisado, el aprendizaje no supervisado y el de aprendizaje de refuerzo.

El aprendizaje supervisado.

El objetivo principal del aprendizaje supervisado es el entrenar un modelo predictivo a partir de datos que han sido previamente etiquetados. Gracias al entrenamiento, el modelo obtendrá la información necesaria para hacer predicciones cuando se le suministre nuevas instancias. El término supervisado, hace referencia a que empleamos un grupo de muestras cuya señal de salida (output) previamente ya conocemos (marcadas o clasificadas).

Dependiendo de la naturaleza de la variable que queremos predecir (variable dependiente) nos enfrentaremos a un problema de clasificación o de regresión. Cuando la variable dependiente o variable a predecir es categórica, se trata de un problema de clasificación, mientras que si la variable es continua estaremos ante un problema de regresión.

En la Figura 1Figura 1, se presenta un esquema que trata de resumir el aprendizaje supervisado.

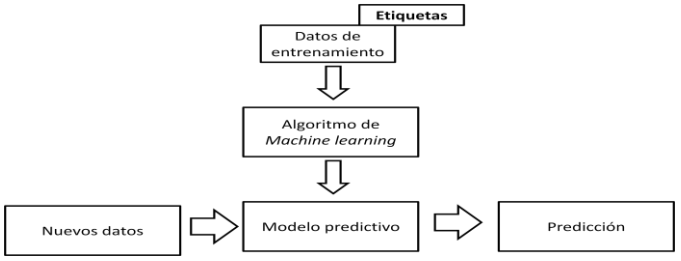

Figura 1: Esquema del aprendizaje supervisado

Empleo de la clasificación para predecir.

La clasificación es un tipo de aprendizaje supervisado donde el objetivo es el predecir etiquetas para

las nuevas instancias. Para clasificar, se emplea el conocimiento adquirido a partir de observaciones pasadas. Las etiquetas son la variable dependiente. El modelo de clasificación más simple es el dicotómico, donde solo hay dos clases. Cuando tenemos más de dos clases, estamos ante un problema de multiclasificación.

El algoritmo aprende un conjunto de patrones que le permiten distinguir entre las posibles clases: sano o enfermo, aceptable o no aceptable, grupo 1, grupo 2 ó grupo 3, etc. Una vez el modelo ha aprendido, predecirá la clase a asignar a las nuevas instancias.

Un ejemplo simple de concepto de clasificación binaria se recoge en la Figura 2 donde 23 casos fueron etiquetados o como positivo (naranja) o como negativo (azul) en un escenario de dos dimensiones (es decir cada muestra tiene dos valores asociados, X1 y X2). Empleando algoritmos de ML supervisado, el algoritmo aprenderá reglas para separar las dos clases, lo que establece un límite de decisión que en la figura se representa como la línea negra central. Cuando entren nuevas muestras sin etiquetas, el algoritmo les otorgará una de las dos etiquetas basándose en los valores X1 y X2 asociados a la nueva muestra.

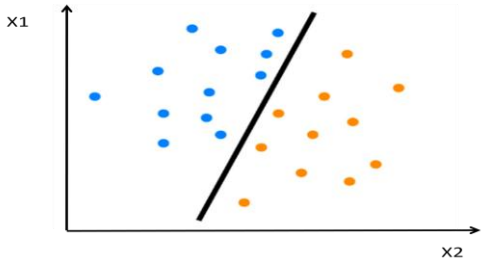

Figura 2: Ejemplo de clasificación

Machine learning y el diagnóstico clínico

Una de las tareas más importantes en el campo de la salud, es el diagnosticar enfermedades (6). En muchos casos, el diagnóstico temprano es muy relevante, no solo para el inicio del tratamiento sino en algunos casos siendo determinante para la supervivencia del paciente.

Los beneficios del ML en la clínica pueden ser múltiples para un sistema de salud sobrecargado. En el campo médico, puede aportar un diagnóstico preciso y rápido de las enfermedades (6), ayudando al clínico a tomar decisiones de una manera más ágil, o proporcionando un apoyo en situaciones donde el diagnóstico es complicado y supone una duda. Para el sistema de salud, esto supone una reducción del coste ya que se reducirían el número de visitas. Y por supuesto habrá un beneficio para el paciente quien se beneficiará de un diagnóstico más precoz y preciso.

ML, *datasets* e investigación

La investigación destina gran cantidad de recursos a la obtención de datos, sin embargo, gracias a las nuevas tecnologías de almacenamiento de datos y a la capacidad para compartirlos, existen

posibilidades para reducir estos costes. Existen grandes *datasets* públicos y privados que almacenan datos que pueden ser empleados por los distintos investigadores para sus estudios.

En el caso del ML, existen numerosos repositorios que almacenan información útil para entrenar algoritmos. Los repositorios más famosos son los de *Kaggle* y UCI *Machine Learning Repository*. Estos repositorios contienen grandes *datasets* que han sido donados por investigadores para que otros grupos hagan uso de ellos.

Muchos de los estudios que han depositado los datos en estos repositorios comparten objetivo, lo que en algunos casos supone que pueden unirse y obtener un *dataset* mayor. Antes de realizar esta unión es necesario estudiarlos para ver si además de objetivo comparten características comunes. En realizar uniones de *datasets* además presenta algunos problemas como son los datos perdidos.

¿Qué son los datos perdidos?

En el mundo de la investigación los valores perdidos es un concepto importante que hay que entender para poder manejar los datos de una manera eficiente. El uso inadecuado de los datos perdidos, puede llevarnos a una inferencia errónea sobre los datos reales que tenemos. Si los manejamos de una manera incorrecta, los resultados obtenidos van a diferir con respecto a aquellos donde no hay datos perdidos presentes (8).

Cuando se identifican datos perdidos, el investigador debe tomar la decisión de dejar los datos tal y como están o la de realizar una imputación de datos para sustituirlos. Esta decisión se toma muchas veces en función del volumen de los datos perdidos, si suponen un bajo porcentaje se suelen eliminar los casos que presentan estos datos perdidos. Sin embargo el eliminar los datos perdidos tiene una serie de repercusiones directas como perder poder estadístico, sesgo de resultados y en algunos casos cambiar los errores estándar y los valores de p en los distintos test estadísticos.

En lugar de eliminar los datos perdidos, otra aproximación que se puede seguir es lo que se conoce como imputación. Existen dos tipos de imputación, la simple y la múltiple. La primera consiste en sustituir los valores perdidos por un valor derivado del resto de los valores presentes de la propia variable (como una media, moda o mediana) mientras que el segundo consiste en emplear técnicas estadísticamente más avanzadas. El empleo de una técnica u otra va a depender fundamentalmente de la naturaleza de los datos perdidos, ya que la imputación múltiple solo la podemos emplear cuando los datos perdidos son de tipo aleatorio *(MAC: Missing completely at random o MAR: Missing at random)* (9).

Con todo ello, en este estudio se busca desarrollar una herramienta que permita distinguir entre pacientes con Parkinson de individuos sanos a partir de datos extraídos a partir de grabaciones de voz.

Materiales y Métodos

UCI *Machine Learning Repository*

El *UCI Machine Learning Repository* es una colección de *datasets* que son usados por la comunidad de ML para el análisis empírico de algoritmos. El repositorio fue creado en 1987 por David Aha y compañeros estudiantes de postgrado en la UC Irvine. Desde entonces, el repositorio ha sido ampliamente empleado por estudiantes, educadores, e investigadores de todo el mundo como una fuente de datos. Como indicador de su impacto, ha sido citado más de 1000 veces, convirtiéndolo en uno de los 100 artículos más citados de las ciencias computacionales. La versión actual del sitio web fue diseñada en 2007 por Arthur Asuncion y David Newman, en colaboración con *Rexa.info en la Universidad de Massachusetts Amherst* (7).

En la actualidad, *UCI Machine Learning Repository* contiene 474 data sets como un servicio para la comunidad del ML. Contiene 9 *datasets* relacionados con la enfermedad de Parkinson. El primero de ellos depositado en 2008 y en 2019 se depositó el último.

A partir de los *datasets* almacenados en este repositorio, se obtuvieron todos los datos posteriormente analizados y empleados para el desarrollo del algoritmo de predicción de Parkinson a partir de grabaciones de voz.

R y Rstudio

R es un ambiente de software libre para la computación estadística y gráfica. El origen de R es un proyecto similar al lenguaje de S. Se puede considerar que R es una implementación de S ya que muchos de los códigos escritos para S también funcionan en R.

R-project proporciona una amplia variedad de técnicas estadísticas y gráficas ampliamente extensibles y extrapolables. R proporciona una ruta *Open Source,* y permite participar en ella de una manera activa. Para el mundo de la estadística y ciencia de datos, R aporta una gran capacidad añadida: la facilidad con la que se producen *plots* (gráficos) bien diseñados y de gran calidad, incluyendo símbolos matemáticos y fórmulas cuando son necesarios.

R no solo proporciona facilidades para realizar gráficos, sino que proporciona un manejo efectivo de los datos y una gran facilidad de almacenamiento. Además permite manejar *arrays*, particularmente en formato de matrices, conteniendo una gran cantidad de herramientas para el análisis de datos. Todo ello hace de R una herramienta ideal para el manejo de los datos y para realizar análisis estadístico de tipo descriptivo e inferencial.

Rstudio es un ambiente integrado de desarrollo para R. El manejo de R como código de texto plano en ocasiones puede ser tedioso, sin embargo *Rstudio* nos proporciona una consola y un editor de sintaxis que apoya directamente la ejecución del código, proporcionando un espacio de trabajo.

En este trabajo, se emplearon R y *Rstudio* para el desarrollo de scripts dedicados al análisis de los

datos que componían los *datasets* y enfocados en última instancia a la toma de decisiones acerca de si los distintos *datasets* podían unirse en uno.

R como herramienta para el análisis descriptivo

Fueron necesarios, además del paquete básico de R, el empleo de diversas librerías enfocadas a la ciencia de datos. Estas permiten un manejo eficiente de los *datasets* y una capacidad extra para visualizar y comparar el contenido de los mismos. Las librerías, son paquetes "extra" que se le añaden a R proporcionándole nuevas capacidades que no tenía y que deben ser cargados de manera explícita.

Las librerías que se hicieron uso enfocadas al análisis descriptivo fueron:

- *xlsx:* destinado a leer y escribir archivos de tipo Excel.
- *stringr*: útil para la limpieza y preparación de los datos. Proporciona un grupo de funciones diseñadas para trabajar con cadenas (*strings*) de la forma más fácil posible.
- *dplyr*: permite la manipulación de los datos de una manera gramatical, proporcionando un grupo consistente de verbos que ayudan a solventar los retos más comunes en la manipulación como seleccionar, filtrar…
- *ggplot2*: paquete enfocado a la creación de gráficos. Se le proporciona a ggplot2 los datos y se le indica como mapear las variables, que gráfico emplear y los detalles que quieres que tenga el gráfico.
- *naniar*: paquete útil para explorar y manejar datos perdidos en los estados iníciales de los análisis. Este paquete proporciona estructuras de los datos y funciones que facilitan la representación de los valores perdidos y examinar las imputaciones.
- *VIM:* Es un paquete útil para la visualización de datos perdidos y/o imputados que han sido introducidos. Esta librería puede ser empleada para explorar los datos y la estructura de los mismos. Dependiendo de la estructura de los datos perdidos, existen métodos que sirven para identificar los mecanismos que los generan y permiten elegir los métodos de imputación apropiados.
- *Hmisc:* contiene funciones para el análisis de datos, gráficas de alto nivel, operaciones utilitarias, funciones para computar el poder y el tamaño de la muestra, importar y anotar *datasets*, imputar valores perdidos, funciones para hacer tablas avanzadas, *clustering* de variables, manipulaciones de cadenas, conversión de elementos de R a *LateX* y código *html*, y guardar variables.
- *corrplot:* es una librería que nos permite realizar matrices de correlación o matrices generales. Contiene algunos algoritmos para reordenar las matrices. Nos permite realizar y ver gráficamente una matriz de correlación entre variables.

Las librerías que se emplearon con un enfoque de análisis inferencial fueron:

- *nortest*: tiene implementado un test de normalidad que permite evaluar la distribución de la muestra.
- *car:* es otro paquete enfocado a la estadística que nos permite entre otras cosas el realizar test de homogeneidad de varianza como el test de *Levene* y el test de *Barlett*.
- *corrplot:* librería que nos permite realizar matrices de correlación o matrices generales. Contiene algunos algoritmos para reordenar las matrices. Nos permite realizar y ver gráficamente una matriz de correlación entre variables.

Existen otros paquetes básicos que se han usado como el *stats* para el análisis de ANOVA y test de *Welch* entre otros.

Análisis estadístico descriptivo e inferencial.

El ANOVA es un test estadístico que se aplica cuando queremos comparar las medias de tres o más grupos. Es un test de contraste de hipótesis que plantea como hipótesis nula que todas las muestras tienen la misma media. Para resolver este contraste de hipótesis, la variabilidad total se reparte entre dos componentes, uno que es explicable por la diferencia entre grupos (varianza entre grupos o *between*), que viene a expresar lo que es el efecto y el segundo componente, el residual que queda dentro de cada grupo (intragrupo, *within*) y que expresa el error (10).

Los requisitos para realizar un ANOVA son:

1. Tipos de variables:
   a. La variable dependiente debe ser cuantitativa.
   b. La variable independiente es la variable de agrupación.
2. Normalidad: La distribución de los residuales debe aproximarse a una distribución normal. Cuando los residuales no superan el test de normalidad, puede haber problemas para aplicar el ANOVA. Aunque este requisito solo es de gran importancia cuando el tamaño muestral de los grupos es menor de 30.
3. Homogeneidad de varianzas: Se mide mediante el test de Levene o el test de Bartlett. Lo ideal es que estos no sean significativos (>0.05) cuando el tamaño muestral de los grupos es menor de 30. Si todos tienen un tamaño superior a 30, la hipótesis aquí exigida, no es importante.

El análisis estadístico, proporciona los argumentos necesarios para tomar la decisión de si unir los *datasets*. Además, permite en caso de ser necesario, discernir qué *datasets* deben eliminarse del proceso y cuáles deben continuar. Por último, proporciona información acerca de qué hacer con las variables (eliminarlas, mantenerlas o imputar datos faltantes). La descriptiva además, nos da información sobre la distribución de la muestra y saber si hay una compensación entre los sanos y los enfermos.

Algoritmos de *Machine learning*

Tras la obtención del *dataset* final, el siguiente paso es el de desarrollar los algoritmos de ML supervisado para poder clasificar a los individuos. Para esta tarea se decidió hacer uso de otro lenguaje de programación: *Python*.

Python

Este es un lenguaje de programación de tipo multiparadigma, que soporta la programación orientada a objetos, programación imperativa, e incluso la programación funcional. *Python* es un lenguaje potente que ha ido creciendo en importancia en la ciencia de datos. Posee una licencia de código abierto siendo dinámico y multiplataforma.

Visualización del *dataset* final:

Antes de desarrollar los algoritmos de ML, es recomendable realizar un proceso de visualización de los datos contenidos en el *dataset* unido para comprender las variables y cómo se comportan.
Al igual que ocurre en R, *Python* puede cargar librerías que le dan una funcionalidad extra no presente en el paquete básico. En este estudio, para la visualización se emplearon las siguientes librerías de *Python*:

- *Pandas*: enfocada a la manipulación y análisis de datos. Ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales proporcionándole a *Python* una versatilidad que de otro modo no tendría.
- *Numpy*: le aporta un mayor soporte para el trabajo con vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con vectores o matrices.
- *Matplotlib*: es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o *arrays*.

Se realizan visualizaciones del tipo de matrices de correlación, histogramas, matrices de diagramas de dispersión, o histogramas de variables combinadas lo que da una representación visual de cómo se comportan las variables y la existencia o no, de correlaciones entre las mismas.

Selección de variables.

Tras ver el comportamiento de las variables, podemos diseñar un modelo de reducción de variables en caso de que sea necesario. Este es un proceso enfocado a disminuir el número de variables aleatorias que entran en los modelos de ML.
Las razones por las que nos puede interesar el reducir las variables son varias:

- identificar y eliminar variables irrelevantes,
- no siempre el mejor modelo es el que más variables incluye,

- mejorar el rendimiento computacional (ahorro en coste y tiempo),
- reducir la complejidad, lo que lleva a facilitar la comprensión del modelo y sus resultados.

Para la reducción de dimensiones se han seguido los siguientes criterios: criterios de dependencia o correlación (información mutua y chi2) con la variable dependiente y el criterio de consistencia (variables redundantes) con las variables independientes. El primero de los métodos, nos muestra cuales son aquellas variables más relacionadas con la variable a predecir y que por tanto tiene más sentido mantener en el modelo y el segundo de los criterios lo que trata es de mostrar si hay si hay variables independientes altamente correlacionadas.

Tratamiento de los datos perdidos:

Para el tratamiento de los datos perdidos se han planteado tres posibilidades. La primera de ellas es la que se conoce como la aproximación de variables completas y que consiste en continuar el análisis solo con aquellas variables que están presentes en todas las muestras. La segunda es la que se conoce como imputación simple que consiste completar los valores perdidos con un valor obtenido a partir del resto de los datos presentes en la variable, como por ejemplo la media. El tercer método planteado es el que se conoce como imputación múltiple y que consiste en realizar la imputación empleando unos modelos más complejos que no solo le da un valor al dato perdido sino que le confiere una cierta capacidad de variabilidad que no confiere el método de imputación simple.

En el estudio, se seguirá como mínimo la aproximación de variables completas y dependiendo de la naturaleza de los valores perdidos se empleará un método de imputación u otro (para emplear la imputación múltiple los valores perdidos tienen que ser de tipo aleatorio, ya sea del tipo missing at random –MAR- o missing completly at random –MCAR-). Se entienden valores como MCAR cuando no hay relación entre si un dato es perdido y cualquier valor del dataset, ya sea perdido u observado (son completamente aleatorios). Por su parte los MAR ocurren cuando la propensión de que se pierda un dato no está relacionada con el resto de valores perdidos, pero sí tiene relación con algunos de los datos observados. En los MAR la pérdida está condicionada por otra variable (11).

Modelos de *Machine learning*

Para el desarrollo de los algoritmos de aprendizaje supervisado se ha empleado *Python* haciendo uso de diversas librerías como *Pandas, Numpy, Matplotlib* y *Scikit-learn*.

- *Scikit-learn (sklearn)*: es una librería de ML aplicado al lenguaje de *Python*. Incluye algoritmos de clasificación, regresión y análisis de grupos. Esta librería nos permitirá entre otras cosas la partición aleatoria del *dataset* en tres partes (entrenamiento, test y validación), y el uso de los modelos de clasificación de Regresión logística (RL), *Random Forest* (RF) y *Support Vector Machine* (SVM) entre otros.

El primer paso para desarrollar los modelos de ML, es dividir el *dataset* de una manera aleatoria en *subsets* (entrenamiento, test y validación) que serán empleados en las distintas fases del desarrollo del modelo. El *subset* de entrenamiento se emplea para entrenar el algoritmo, el *subset* de validación se emplea para analizar los resultados del algoritmo, para en caso de ser necesario ajustar los *hiperparámetros* del modelo. El *subset* de test, se emplea para validar el modelo predictivo.

No siempre se puede disponer de *subset* de test, ya que dependerá del tamaño de la muestra disponible para crear el modelo. Si el tamaño de la muestra es pequeño, se emplean otros métodos de validación, como la validación cruzada. Esta hace uso de subconjuntos de los datos disponibles para realizar el entrenamiento del modelo y posteriormente validarlo (12). Dentro de la validación cruzada existen distintos métodos como son el *k-fold cross-validation* o el *leave-one-out cross-validation* (LOOCV) entre otros.

En caso de disponer del tamaño suficiente los tres *subsets*, el *subset* de test, no pude participar en la creación del modelo. Es decir, debe actuar como una muestra de datos externa. Por ello, la partición de los datos en los distintitos *subsets* debe hacerse lo primero.

No hay una regla estándar para hacer el tamaño de las particiones. En este estudio, en caso de ser viable por tamaño muestral, se hará una partición con una proporción 60% para el *subset* de entrenamiento, 20% para el *subset* de test y 20% para el *subset* de validación.

Algoritmos de ML para clasificación supervisada:

Son múltiples los algoritmos empleados en el ML:

- Regresión logística: La regresión lineal trata de establecer un modelo para la relación entre un cierto número de características y una variable objetivo continúa ajustándose a una línea. El algoritmo trata de minimizar el coste de una función de error cuadrático. Los coeficientes se corresponden con la recta óptima. Cuando analizamos una variable continua empleamos la regresión lineal, sin embargo ante problemas en los que la variable dependiente es categórica, empleamos la regresión logística. Este método es útil para modelar la probabilidad de que ocurra un evento en función de una serie de variables independientes.
- *Support Vector Machines* (SVM): La mayoría de los métodos de aprendizaje se centran en minimizar los errores cometidos por el modelo generado a partir de los ejemplos de entrenamiento, sin embargo SVM trata de minimizar lo que se conoce como riesgo estructural. La idea fundamental que se esconde tras SVM es la de seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase para, de esta forma, conseguir lo que se denomina un margen máximo a cada lado del hiperplano (13). A la hora de definir el hiperplano, sólo se consideran aquellos valores de

cada clase que caen justo en la frontera de dichos márgenes, siendo lo que se conoce como vectores de soporte.

- *Random Forest*: El *Random forest*, está formado por muchos árboles de decisión. Cada uno de los árboles es creado haciendo uso de un *subset* de atributos originales, con los que clasifica la población a estudio. Los árboles "votan" sobre como clasificar una nueva instancia, y el algoritmo de *random forest* hace un *boostrap* de los votos para seleccionar el mejor predictor (14). Esto es muy útil para evitar lo que se conoce como *overfitting* o sobre entrenamiento. El *bootstrap* es un método de re-muestreo y se emplea para cuantificar la incertidumbre asociada con un estimador estadístico. Trabaja mediante el re-muestreo N veces con reemplazo desde el conjunto de entrenamiento para formar nuevas tablas de datos (Bootstraps). El modelo se estima en cada una de estas nuevas tablas (*bootstraps*) y las predicciones se hacen para la tabla original de datos o conjunto de entrenamiento. Este proceso se repite muchas veces y posteriormente se promedian los resultados.

En este estudio tras analizar el problema que se tenía entre manos se optó por emplear estos tres modelos ya que estos modelos cumplen con las necesidades del objetivo del proyecto.

Resultados

Análisis de los *datasets* relacionados con Parkinson.

Tras analizar la descripción y la composición de los 9 *datasets* localizados inicialmente en el repositorio de UCI *Machine Learning Repository*, solamente 5 fueron seleccionados para los análisis posteriores. Los 4 restantes fueron eliminados porque no estaban relacionados con datos obtenidos a partir de la voz o porque no se podía acceder a los datos.

Los 5 datasets que finalmente entraron en el estudio fueron:

- Little et al. (15)
- Tsanas et al. (16)
- Kursun et al. (17)
- Sakar et al. (18)
- Pérez et al. (19)

Todos los *datasets* se habían empleado para clasificar entre sano o pacientes con Parkinson (variable dependiente) haciendo uso de variables relacionadas con la voz. Sin embargo, el análisis descriptivo mostró que había diferencias entre ellos.

Dado que el trabajo de Little et al. (15) fue el primero en depositarse, éste fue el empleado como molde para crear el dataset más grande. Este *dataset* presentaba 24 variables, algunas de ellas descartables al tratarse de los IDs de los participantes del estudio. De las otras 23 variables, la variable dependiente era el *status* (sanos = 0; pacientes = 1). El resto eran variables relacionadas con la voz. El conjunto de variables se recogen en la Tabla 1. Este *dataset* contenía datos asociados de 195 mediciones.

El resto de los *datasets* tenían distinto número de variables y mediciones:

- Tsanas -> 23 variables y 5875 mediciones.
- Kursun -> 28 variables y 1208 mediciones.
- Sakar -> 755 variables y 756 mediciones.
- Pérez -> 48 variables y 240 mediciones.

Se analizó si las variables contenidas en el estudio de Little, también habían sido estudiadas en el resto de los estudios. Tras realizar el análisis comprobamos que no todos los estudios contemplaban todas esas variables (Tabla 1).

Variables	Dataset				
	Little	Tsanas	Sakar	Kurson	Pérez
name	Sí	Sí	Sí	Sí	Sí
MDVP.Fo.Hz	Sí	No	No	No	No
MDVP.Fhi.Hz.	Sí	No	No	No	No
MDVP.Flo.Hz.	Sí	No	No	No	No
MDVP.Jitter	Sí	Sí	Sí	Sí	Sí
MDVP.Jitter.Abs.	Sí	Sí	Sí	Sí	Sí
MDVP.RAP	Sí	Sí	Sí	Sí	Sí
MDVP.PPQ	Sí	Sí	Sí	Sí	Sí
Jitter.DDP	Sí	Sí	Sí	Sí	No
MDVP.Shimmer	Sí	Sí	Sí	Sí	Sí
MDVP.Shimmer.dB	Sí	Sí	Sí	Sí	Sí
Shimmer.APQ3	Sí	Sí	Sí	Sí	Sí
Shimmer.APQ5	Sí	Sí	Sí	Sí	Sí
MDVP.APQ	Sí	Sí	Sí	Sí	Sí
Shimmer.DDA	Sí	Sí	Sí	Sí	No
NHR	Sí	Sí	No	Sí	No
HNR	Sí	Sí	No	Sí	No
status	Sí	Todos pacientes	Sí	Sí	Sí
RPDE	Sí	Sí	Sí	No	Sí
DFA	Sí	Sí	Sí	No	Sí
spread1	Sí	No	No	No	No
spread2	Sí	No	No	No	No
D2	Sí	No	No	No	No
PPE	Sí	Sí	Sí	No	Sí

Tabla 1: Distribución de variables en los distintos *datasets.*

Como se puede observar, había una gran variación tanto en el número de mediciones hechas en cada estudio como en el número de variables. Para nuestro análisis, cada medición se consideró como un paciente individual (aunque en la mayoría de los casos eran mediciones hechas del mismo paciente repetidas veces).

Todas las variables que no habían sido estudiadas en el estudio de Little et al fueron eliminadas de análisis posteriores. Aquellas variables que no estaban presentes en todos los *datasets* fueron analizadas para decidir qué hacer con ellas.

Como muestra en la Tabla 1, 12 variables están contenidas en todos los *datasets*, 5 están presentes en 3 de ellos, y 6 solo están presentes en el *dataset* de Little. Dado que el objetivo de este estudio era el de juntar diversos *datasets*, aquellas variables que solo estaban presentes en el trabajo de Little et al., también fueron eliminadas trabajando con las restantes. Antes de tomar más decisiones acerca de las variables incluidas en el *dataset* final, se hizo un análisis estadístico para comprobar cómo se comportaban las distintas variables en los distintos *dataset*s y valorar si era factible unirlos en uno.

Análisis estadístico de las variables contenidas en los datasets.

Para comprobar si existían diferencias estadísticas entre las distintas variables de los *datasets*, se realizó un análisis de contraste de hipótesis. Al analizarse más de dos grupos (análisis de k medias), es test empleado para el contraste de hipótesis es el ANOVA de una vía. Este análisis se realizó a varios niveles, primero de toda la muestra y después segregando por condición de estudio (sanos y enfermos).

El test de ANOVA, responde a la pregunta de si hay diferencias en la media de alguno de los grupos a comparar, es decir que se comparan todas las medias. La comparación de medias que hace ANOVA es mediante el estudio de varianzas.

Previamente a realizar el análisis de ANOVA, hubo que preparar la base de datos para su análisis. Para hacer este tipo de análisis, tiene que tener un formato conocido como '*long'* donde cada columna es una variable y cada fila un paciente. Además debemos tener una columna que identifique a cada paciente con su *dataset* correspondiente y que será tratada como un factor. En este tipo de análisis además, no puede haber NAs (*Not Available* o dato vacio), por lo que en aquellas variables que estaban presentes solo en alguno de los *dataset* hay que hacer un análisis separado.

Se comprobaron los criterios necesarios para realizar el análisis de ANOVA: tipo de variable, normalidad y homogeneidad de varianza.

El criterio de los tipos de variables se cumplían en todos los casos.

El criterio de normalidad fue evaluado tanto visualmente (histogramas y normal q-q *plot*) como mediante test específicos (al tratarse de una muestra mayor de 50 se empleó Kolmogorov-Smirnov).

El criterio de homogeneidad de varianza se evaluó visualmente mediante box *plot* y mediante test específicos (test de Levene y test de Barlett).

Todos los test fueron significativos mostrando que las muestras no se distribuían de manera normal ni existía homogeneidad de varianza.

Dado que todas las muestras presentaban tamaños superiores a 30, se podía haber tomado la decisión de realizar solamente un test de ANOVA, sin embargo dados los resultados de los test de normalidad y homogeneidad de varianza, se decidió realizar además un test de ANOVA heterodástico o test de Welch. Todos las variables analizadas mostraron significación estadística ($p< 0.05$) tanto para el ANOVA como para el test de Welch. Este tipo de test, contrasta la hipótesis de que todas las medias son iguales, por lo que un resultado significativo indica que al menos uno de los grupos, tiene una media diferente, sin embargo no sabemos cuál es el grupo diferente.

Para responder a la pregunta de qué grupo difiere del resto, hay que realizar un test Post Hoc mediante un test de comparaciones múltiples. En los test de comparaciones múltiples, los niveles

de significancia deben ser ajustados en función del número de comparaciones. En este estudio, se realizaron dos test post-hoc, el primero de ellos un test de comparación de medias ajustado por el método de Holm y el segundo de los test fue un Tukey HSD.

A modo de ejemplo, a continuación presento los resultados obtenidos de la variable Jitter.

Para el análisis visual de normalidad, se realizó un histograma de la distribución de la variable junto con su curva de normalidad teórica para cada uno de los datasets (Figura 3). En la imagen se observa como la variable no sigue una distribución normal en ninguno de los *dataset.* Lo mismo ocurre cuando observamos el gráfico q-q (Figura 5) donde se ve como las muestras se desvían de la línea indicativa de la normalidad. En la Figura 4 se presentan los resultados obtenidos de los test de normalidad (Kolmogorov-Smirnov) en cada uno de los *dataset*.

La distribución de la variable se muestra en la representación de los diagramas de cajas (Figura 7), en los que se observa como existen dos *dataset* (Kursun y Pérez) en los que esta variable se distribuye de una manera diferente al resto. El test estadístico que evalúan la homogeneidad de varianza (test de Barlett) confirma lo sospechado visualmente, ya que mostró significación estadística ($p<0.05$), como se muestra en la Figura 6.

Tras evaluar la normalidad y la homogeneidad de varianza, se realizaron los test de contraste de hipótesis ANOVA y test de Welch. Ambos mostraron que había diferencias estadísticamente significativas entre alguna de las medias de los *datasets* (Figuras 8). Como se muestra en las figuras Figura 9 y Figura 10 los test post hoc mostraron que el *dataset* de Kursun difería del resto, lo mismo ocurría con el *dataset* de Pérez. Entre el resto de los *dataset* no había diferencias estadísticamente significativas.

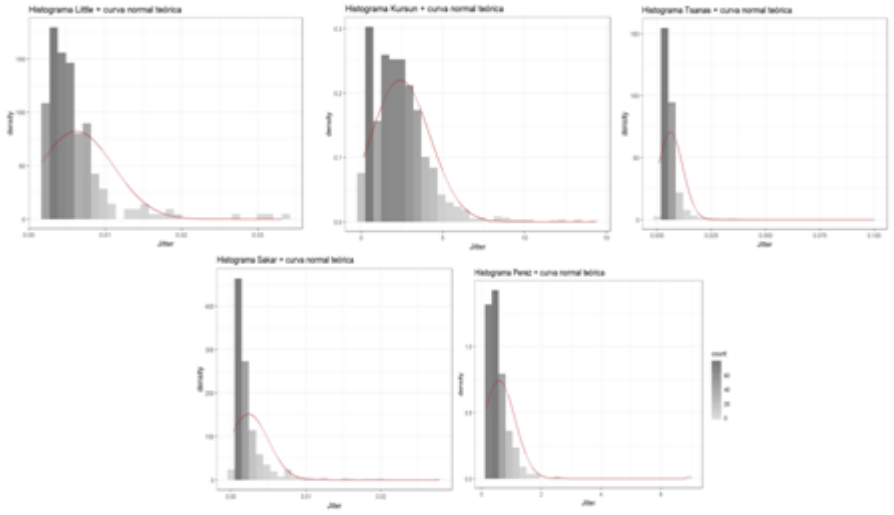

Figura 3: Histograma con línea de normalidad teórica de la variable Jitter para cada uno de los dataset.

```
dataset_ANOVAS$Dataset: little

 Lilliefors (Kolmogorov-Smirnov) normality test

data:  x$Jitter
D = 0.18682, p-value < 2.2e-16

dataset_ANOVAS$Dataset: kursun

 Lilliefors (Kolmogorov-Smirnov) normality te

data:  x$Jitter
D = 0.10412, p-value < 2.2e-16

dataset_ANOVAS$Dataset: tsanas

 Lilliefors (Kolmogorov-Smirnov) normality test

data:  x$Jitter
D = 0.21108, p-value < 2.2e-16

dataset_ANOVAS$Dataset: sakar

 Lilliefors (Kolmogorov-Smirnov) normality test

data:  x$Jitter
D = 0.22563, p-value < 2.2e-16

dataset_ANOVAS$Dataset: perez

 Lilliefors (Kolmogorov-Smirnov) normality test

data:  x$Jitter
D = 0.2079, p-value < 2.2e-16
```

Figura 4: Resultados de los test de normalidad para la variable Jitter en cada uno de los datasets.

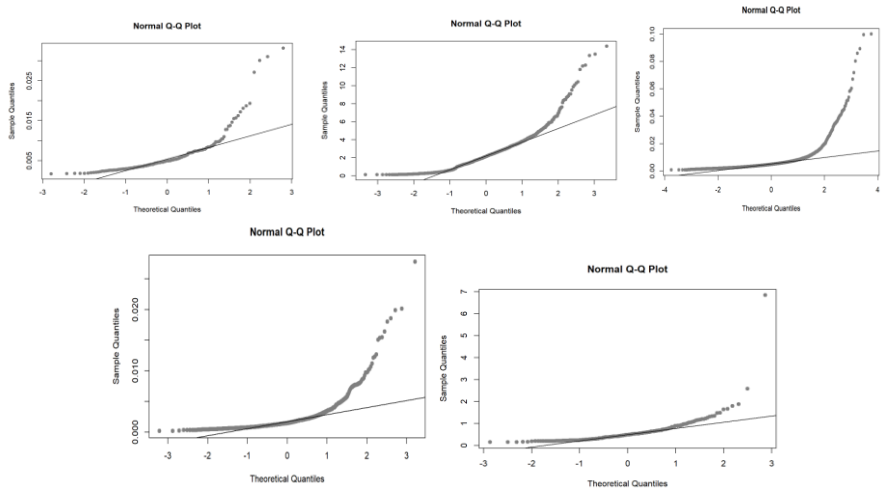

Figura 3: Q-Q plot de la variable Jitter para cada uno de los datasets. Arriba a la iquierda Little, arriba a la derecha Kursun, en medio a la izquierda Tsanas, en medio a la derecha Sakar y abajo Pérez.

```
	Bartlett test of homogeneity of variances

data:  dataset_ANOVAS$Jitter by dataset_ANOVAS$Dataset
Bartlett's K-squared = 64744, df = 4, p-value < 2.2e-16
```

Figura 6:Test de homogeneidad de varianza entre los distintos datasets para la variable Jitter.

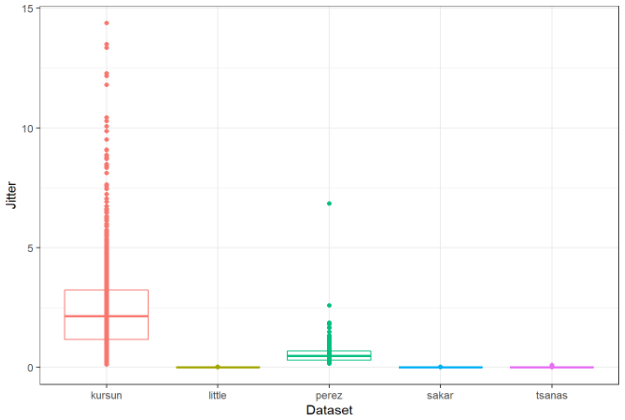

Figura 7: Representación de boxplot de la distribución de la variable Jitter para cada uno de los dataset.

```
                          Df Sum Sq Mean Sq F value Pr(>F)
dataset_ANOVAS$Dataset     4   5875  1468.8    3021 <2e-16 ***
Residuals               8269   4020     0.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


	One-way analysis of means (not assuming equal variances)

data:  dataset_ANOVAS$Jitter and dataset_ANOVAS$Dataset
F = 852.27, num df = 4.00, denom df = 765.09, p-value < 2.2e-16
```

Figura 8: Resultado del ANOVA (arriba) y Test de Welch (abajo) para la variable Jitter.

```
 Pairwise comparisons using t tests with pooled SD

data:  dataset_ANOVAS$Jitter and dataset_ANOVAS$Dataset

       kursun little perez  sakar
little <2e-16 -      -      -
perez  <2e-16 <2e-16 -      -
sakar  <2e-16 1      <2e-16 -
tsanas <2e-16 1      <2e-16 1

P value adjustment method: holm
```

Figura 9: Test de comparación de medias con ajuste de Holm para la variable Jitter.

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = dataset_ANOVAS$Jitter ~ dataset_ANOVAS$Dataset)

$`dataset_ANOVAS$Dataset`
                       diff         lwr         upr     p adj
little-kursun -2.389755e+00 -2.53657783 -2.24293257 0.0000000
perez-kursun  -1.811989e+00 -1.94643855 -1.67753898 0.0000000
sakar-kursun  -2.393651e+00 -2.48187616 -2.30542631 0.0000000
tsanas-kursun -2.389822e+00 -2.44992352 -2.32972028 0.0000000
perez-little   5.777664e-01  0.39435068  0.76118219 0.0000000
sakar-little  -3.896035e-03 -0.15669731  0.14890524 0.9999947
tsanas-little -6.669984e-05 -0.13854701  0.13841361 1.0000000
sakar-perez   -5.816625e-01 -0.72261669 -0.44070825 0.0000000
tsanas-perez  -5.778331e-01 -0.70311944 -0.45254683 0.0000000
tsanas-sakar   3.829335e-03 -0.06967951  0.07733818 0.9999084
```

Figura10: Test de Tukey con ajuste HDS para la variable Jitter.

Al existir diferencias estadísticamente entre los distintos *datasets*, se realizó un pequeño análisis más en profundidad de la variable. En el caso de la muestra de Pérez, la bibliografía reveló que la variable estaba en otra escala (había que dividir el valor entre 100), y una vez transformada el análisis estadístico indicó que no había diferencias significativas con el resto de *datasets*. Sin embargo en el caso de la muestra de Kursun, el análisis no abrió la puerta a ningún cambio y por lo tanto difería del resto de los *dataset* en la variable.

El mismo proceso se realizó para el resto de las variables. En la Tabla 2 se presentan los resultados de la comparación de variables empleando toda la muestra (sin distinguir entre sanos y enfermos). En rojo se muestran aquellas variables con diferencias significativas entre los *datasets*. En conjunto, el *dataset* de Kursun difería del resto de una manera significativa.

Lo mismo sucedió cuando segregamos por condición de sanos y enfermos (no se muestran los resultados). Finalmente, se tomó la decisión de eliminar los datos de Kursun ya que parecía inviable el juntarlos con el resto.

Toda la muestra					
Variable	**Little**	**Kursun**	**Sakar**	**Tsanas**	**Pérez**
Jitter					
ABS_Jitter					
RAP					
PPQ					
Shimmer					
ShimmerDB					
Shimmer_APQ3					
Shimmer_APQ5					
Shimmer_APQ11					
JitterDDP					
Shimmer_DDA					
RPDE					
DFA					
PPE					
NHR					
HNR					

Tabla 2: Diferencias entre los distintos *datasets* según variables cuando analizamos toda la muestra (sin distinguir entre sano y enfermos). En rojo se representa cuando hay diferencias estadísticamente significativas en los test post-hoc, en amarillo-naranja los *dataset* que no mostraron diferencias.

Unión de los *datasets*

Una vez realizado el análisis estadístico, fueron 4 *datasets* que fueron incluidos en los modelos de ML: Little, Sakar, Tsanas y Pérez.

Al unirse, se consiguió un tamaño muestral de 7006 instancias aunque distribuidas de una manera descompensada. 360 eran mediciones de individuos sanos y 6706 de pacientes con Parkinson. Dada esta descompensación, se tomó la decisión de hacer un remuestreo aleatorio sin reemplazo de las muestras de pacientes. De esta forma, el *dataset* final consistió en 860 mediciones distribuidas en 360 sanos y 500 pacientes.

Dado que había variables no presentes en todos los *datasets*, se tomó la decisión de hacer dos análisis:

1. Análisis solo con aquellas variables presentes en todas las instancias
2. Imputar los datos de las variables faltantes. Dado que no estábamos ante datos perdidos de tipo *random*, se decidió hacer imputación simple. A todos aquellos valores vacíos, se les asignó el valor promedio de esa variable empleando para el cálculo todas aquellas instancias de las que se disponía.

Teniendo estos dos conjuntos de datos se crearon modelos de ML para cada uno de ellos. Finalmente nos quedaríamos con el algoritmo de predicción que mejores resultados presentase. Ambos *dataset* estaban compuestos por las mismas instancias ya que al realizar la selección de las muestras, se fijó la semilla en el mismo valor. Lo mismo se hizo que al realizar el *subsampling* posterior para obtener el mismo *subset* de entrenamiento, test y validación.

Análisis previo al ML

Este análisis se hizo para las dos aproximaciones (solo datos completos e imputación simple), sin embargo aquí se presentan solo los resultados del análisis con datos completos.

Lo primero que se hizo antes de modelar los algoritmos de ML, fue observar cómo se comportaban las variables para en caso de ser necesario, realizar un proceso de reducción de variables.

El análisis de las variables se realizó de una manera principalmente visual mediante histogramas en los que se representaban en un color los individuos sanos (azul) y en otro los individuos con Parkinson (rojo). Este análisis visual nos permite observar cómo se comporta la variable dependiente en función de una variable independiente concreta. En la Figura 11 podemos ver cómo se distribuye la variable dependiente en función de la variable independiente Jitter. Visualmente esta variable no es capaz de separar los grupos de la variable dependiente, sin embargo puede ser útil en combinación con otras variables.

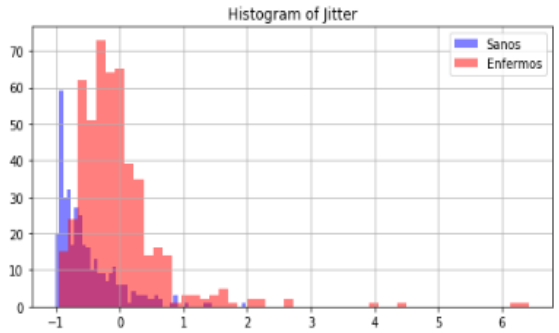

Figura 11: Representación del Status sano/Parkinson en función de la variable Jitter.

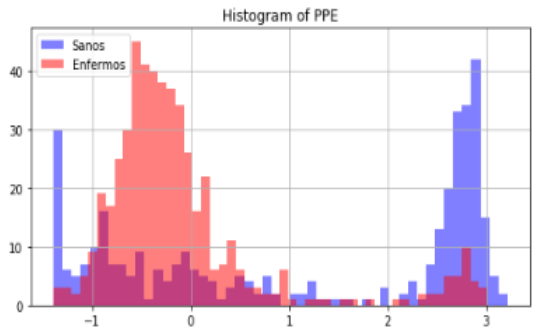

Figura 12: Representación del Status sano/ Parkinson en función de la variable PPE.

Cuando representamos la variable PPE, la representación nos indica que esta variable puede separar bastante bien las clases, como se observa en la Figura 12, donde se ve como la mayoría de los enfermos se sitúan en valores de RPDE alrededor de 0.2 y la mayoría de los sanos alrededor de un valor de RPDE de 0.8.

En relación a una posible necesidad de reducir variables se hicieron otro tipo de análisis como análisis de correlación de las variables independientes y análisis de correlación o dependencia con respecto a la variable dependiente.

En los análisis de dependencia emplearon criterios de información mutua y de chi2, observándose como las variables independientes que más estaban asociadas con la dependiente eran RPDE y PPE para la información mutua y el ShimmerDB y PPE en el análisis de chi2. En la Figura 13 se muestran las representaciones mediante *scatter plot* de la variable dependiente empleando las variables RPDE y PPE como dimensiones y de las variables ShimmerDB y PPE.

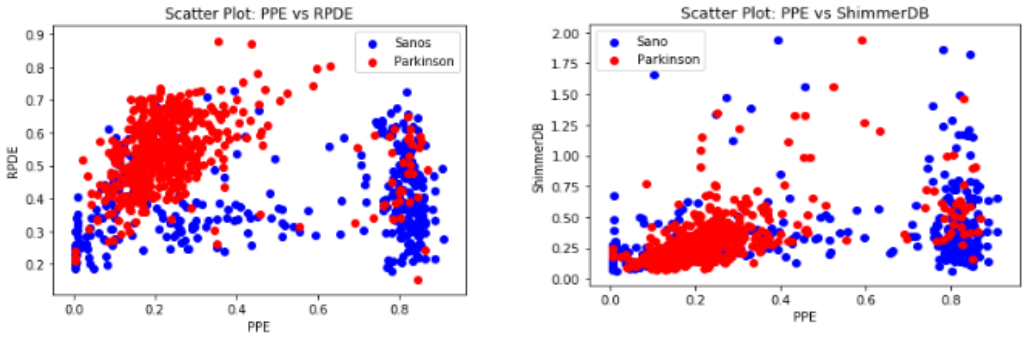

Figura 13: Representación scatter plot de la variable status en función de las variables RPDE y PPE (Izquierda). Representación scatter plot de la variable status en función de las variables ShimmerDB y PPE.

El análisis de correlación entre las variables independientes sirvió para ver que variables estaban más asociadas entre sí. El saber que variables están correlacionadas, nos sirve para en caso de tener que eliminar variables, poder decidir de una manera más fácil cual eliminar. En la Figura 14 se muestra la matriz de correlación entre las variables independientes.

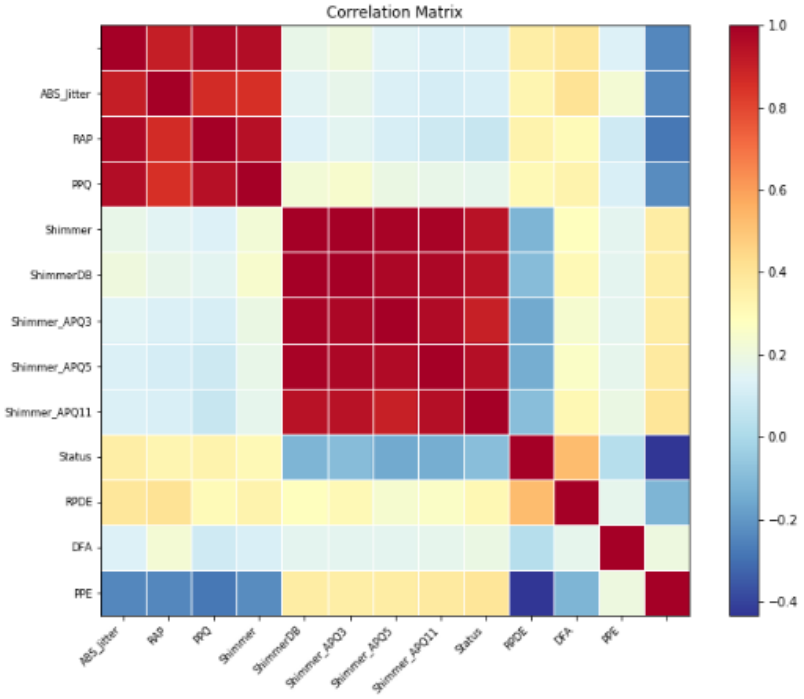

Figura 14: Matriz de correlación de las variables.

Este mismo análisis se realizó para la muestra en la que se habían hecho imputaciones simples.

Modelo de Machine learning

Una vez acabados los análisis previos, se procedió a modelar los algoritmos de ML: Regresión logística, SVM y *Random forest*. Estos modelos fueron estudiados tanto para el caso en el que ha seguido la aproximación de datos completos y en los que se ha realizado imputación simple. Para la selección de la mejor aproximación y mejor modelo de ML, se emplearon distintas métricas:

- Matriz de confusión
- Valor predictivo positivo y negativo
- F1-score
- Sensibilidad del modelo y especificidad
- *Likelihood Ratio* positivo y *Likelihood Ratio* negative
- Exactitud del modelo y AUC

Para validar el modelo de ML, se emplearon dos métodos de validación cruzada y el de validación mediante una muestra independiente. En ambos métodos, se midió la exactitud del modelo. Esto nos permite evaluar si el modelo está sobre-aprendido (*overfitting*), si es necesario reducir las variables que entran en el modelo y valorar como se comportará el algoritmo ante nuevas muestras.

	Regresión logística		**SVM**		***Random forest***	
	Disp.	**I. Simple**	**Disp.**	**I. Simple**	**Disp.**	**I. Simple**
Valor predictivo positivo	0.855	0.854	0.9	0.834	0.881	0.909
Valor predictivo negativo	0.823	0.811	0.847	0.841	0.83	0.849
Sensibilidad	0.881	0.871	0.891	0.9	0.881	0.891
especificidad	0.788	0.788	0.859	0.746	0.83	0.873
f1-score	0.84	0.84	0.87	0.84	0.86	0.88
***Likelihood Ratio* Positivo**	4.17	4.124	6.326	3.553	5.213	7.029
***Likelihood Ratio* Negativo**	0.239	0.242	0.158	0.281	0.191	0.142
Exactitud	0.843	0.837	0.872	0.837	0.86	0.883
AUC	0.887	0.892	0.918	0.889	0.942	0.956

Tabla 3: Resumen de los resultados obtendos en los distintos modelos. Disp.= Datos disponibles, I. Simple = Imputación Simple.

En la Tabla 3 se recogen los resultados obtenidos a partir de los distintos modelos de ML. El modelo que mejores resultados tuvo fue el *Random forest,* aplicado a la muestra en la que se había realizado imputación simple*,* con los siguientes hiperparámetros:

- Bootstrap = True; class_weight = None; criterion = 'entropy' ; max_depth=None; max_features = sqrt(n_features); max_leaf_node = None; min_impurity_decrease = 0.0; min_impurity_split = None; min_sample_leaf = 1; min_sample_split = 2; min_weight_fraction_leaf = 0.0; n_etimators = 500; n_jobs = 1; oob_score = False; random_state = 1; verbose = 0; warm_start = False

Este modelo obtuvo una predicción con 152 aciertos y 20 errores. La matriz de confusión obtenida a partir de este modelo, junto con la curva de predicción se recogen en la Figura 15.

Real	0	1
Prediccion		
0	62	11
1	9	90

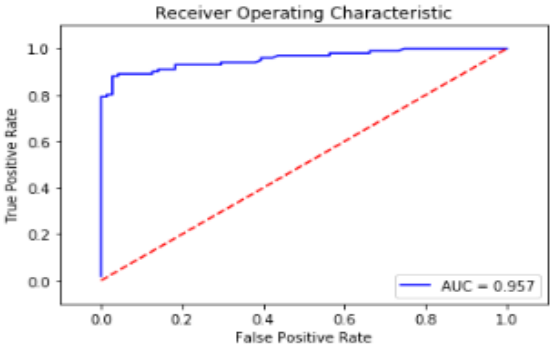

Figura 15: Matriz de confusión obtenida del modelo de *Random forest* (Izquierda). Representación de la curva de acierto del modelo (derecha).

El resultado de la exactitud media del modelo en la validación cruzada fue de 0.866 (Al realizarse una validación cruzada 10 *folds* se obtienen 10 resultados, el resultado que se presenta es el promedio). En el caso de la validación con una muestra independiente (*subset* de validación), se obtuvo un valor de exactitud del modelo de 0.907.

Se hicieron pruebas con reducción de variables pero no hubo mejora en los resultados obtenidos.

Discusión

Repositorios y *dataset*.

En la investigación convencional, gran parte del dinero destinado a los estudios va enfocado a la obtención de datos. Sin embargo, hoy en día existen herramientas y recursos que permiten el reciclaje de datos obtenidos de otros estudios. La tendencia actual de la ciencia es la de compartir estos datos que permiten por un lado replicar los estudios hechos por otros grupos y por otro lado el emplear datos ya obtenidos para realizar estudios de formación o para estudios piloto que permitirán justificar estudios posteriores más grandes. Ejemplos de ello son bases de datos como GEO (Gene Expression Omnibus) o el repositorio que he empleado en este estudio: UCI Machine Learning repository.

Estos repositorios contienen grandes cantidades de *datasets* cuyos objetivos y metodologías son muy diferentes. Por ello, a la hora ser empleados es necesario documentarse previamente a cerca de los datos que contienen y de cómo han sido obtenidos. El que existan múltiples *datasets* enfocados al mismo problema, nos brinda la oportunidad de unirlos en uno más grande, ya que no lo olvidemos, gran parte de la potencia estadística de un estudio recaerá en el tamaño muestral. Por lo tanto, esta aproximación de reciclaje de datos nos permite llegar a una potencia estadística superior que la obtenida por los estudios individuales.

Análisis estadístico

Esta aproximación puede resultar atractiva a primera vista, sin embargo antes de poder unir los distintos grupos de datos, es necesario aplicar un análisis que justifique o indique que no hay diferencias entre las distintas variables obtenidas (lamentablemente, en algunos estudios y publicaciones, la metodología seguida para la obtención de los datos no siempre es tan clara o explicativa como debería ser). En este estudio, se ha seguido una metodología basada en la estadística descriptiva e inferencial, enfocada a demostrar que las variables contenidas en los distintos estudios no diferían unos de otros. Esta información estadística, proporciona al investigador de información necesaria para tomar decisiones del tipo que *datasets* se pueden integrar o que variables son integrables (cuando no hay diferencias estadísticamente significativas). En los casos que haya diferencias estadísticas en las variables, le puede indicar al investigador en que variable debe profundizar la investigación para decidir si es viable hacer algún tratamiento que le acerque a los datos del resto de los *dataset*, y en el peor de los casos, el justificar la eliminación de esa variable del estudio posterior. En el presente estudio, esta aproximación estadística ha llevado a la eliminación de un *dataset* completo y a la eliminación de 6 variables.

Otra de las repercusiones que tiene el análisis estadístico, es que devuelve conocimiento acerca de datos perdidos. Este conocimiento no solo es que datos faltan, sino cómo se comportan esos datos. Este conocimiento es necesario para tomar la decisión sobre cómo tratar esos datos perdidos. En este estudio, el análisis nos mostró que los datos perdidos no eran de tipo *random* y que por lo tanto la

aproximación de imputación múltiple no era factible y por lo tanto se podía seguir el camino del trabajo con datos disponibles o el de imputación simple.

Finalmente, otro de los aportes que tiene el estudio estadístico previo, es el de conocer la correlación de las variables independientes con el resto de las variables independientes y con la variable dependiente. Cuando una variable independiente posee alta correlación con otra u otras puede ser debido a una combinación lineal de alguna de ellas, esto es lo que se llama multicolinealidad. La repercusión que tiene la colinealidad en los modelos es que generan modelos con muy poco poder explicativo o de difícil interpretación, lo que lleva a dificultar la interpretación de la importancia de cada una de las variables independientes en el modelo. Los modelos de ML del tipo *Random forest* o SVM, son modelos que de por sí no permiten una fácil trazabilidad de la importancia de las variables que han entrado en el modelo. Por ello, a pesar de que algunas de las variables independientes mostraban correlación, esta información no ha sido empleada para eliminar variables a priori. Esta información junto con las correlaciones con la variable dependiente, iba enfocada a que en el caso de ser necesario eliminar variables tener información sobre que variables debían ser eliminadas y cuáles no.

Modelos de Machine learning.

El aprendizaje automático o *Machine learning*, es una rama de la inteligencia artificial (IA) que trata de implementar algoritmos matemáticos enfocados al conocimiento tanto a nivel de adquisición como a la acumulación y mejora del mismo a partir de datos y tratando de imitar el razonamiento humano (20). La base de este conocimiento son los datos suministrados a los algoritmos. La IA encontrará patrones, reglas y asociaciones que finalmente derivan al conocimiento. El hecho de que el conocimiento deriva de los datos tiene una implicación muy importante y es que la calidad del conocimiento dependerá de la calidad de los datos que se le suministre y de la calidad del propio algoritmo.

En el campo de la medicina, ya se han ido desarrollando numerosas herramientas cuya base fundamental es la IA y que sirven de apoyo para los clínicos. Prueba del auge del ML en el mundo sanitario, es el creciente número de publicaciones científicas que tienen como tema central el ML. En las que se trata de extraer información de los datos obtenidos de pacientes que han participado en estudios, para que cuando nuevos pacientes acudan a consultas se les pueda realizar predicciones. Los modelos de ML soportan todo tipo de datos, y dependiendo de si la variable respuesta es conocida o no, estaremos ante un problema de aprendizaje supervisado o no supervisado respectivamente.

En nuestro caso hemos estado ante un problema de aprendizaje supervisado, donde teníamos participantes etiquetados como sanos o enfermos con Parkinson. Este tipo de aprendizaje genera conocimiento a partir de datos en los que conocemos la variable a predecir (21).

El esquema del aprendizaje supervisado se puede resumir de la siguiente forma:

Datos (ejemplos con variable respuesta conocida) -> **Aprendizaje** (se entrenan los modelos con los

datos) -> **Modelo de predicción** (Se genera el mejor de los modelos de predicción) -> **Conocimiento** (El modelo tiene todo lo necesario para asociar una variable respuesta a las nuevas instancias) -> **Predicción** (Ese modelo devuelve la variable respuesta ante nuevos pacientes).

Dentro del aprendizaje supervisado y dependiendo de las características de la variable dependiente (variable respuesta) vamos a tener distintos tipos de modelos: regresión o clasificación. En este estudio, la variable dependiente era una variable categórica, por lo que necesitábamos un modelo enfocado a la clasificación, más concretamente ante un problema de clasificación dicotómica.

Para solventar el problema de clasificación existen múltiples técnicas como son Regresión logística (22), Árboles de decisión (23), *Random forest* (24), Máquinas de vestor soporte (SVM) (25), Redes neuronales (26), K-vecinos más cercanos (23). Cada modelo presenta sus ventajas e inconvenientes. En el presente estudio se decidió emplear tres de ellos, la regresión logística, SVM y *Random Forest*.

De todos los modelos estudiados, el que mejor resultado dio fue el *Random forest* con imputación simple devolviendo el mejor de los resultados en todos los parámetros valorados. Al realizar la validación del modelo, se obtuvieron valores muy buenos tanto en la validación cruzada (0.866) como en la validación con una muestra independiente (0.907), lo que es indicativo de que el modelo no está sobre aprendido y puede ser extrapolable a otras muestras.

Puede llamar la atención que en el caso del SVM (Tabla 3) los valores obtenidos en el modelo empeoran sensiblemente cuando empleamos la aproximación de la imputación simple. La razón de este suceso es el hecho de que cuando trabajamos con datos disponibles, fue posible emplear el *kernel* polinómico, mientras que cuando empleamos la imputación simple, el ordenador no pudo con el *kernel* polinómico (problemas posiblemente de coste computacional) y los resultados obtenidos son para el kernel lineal. Cabe esperar el que si se hubiese podido emplear el *kernel* polinómico, el modelo de SVM mejorase al de *Random forest*.

Este estudio ha mostrado muy buenos resultados a la hora de emplear el ML con pruebas no invasivas destinadas a la clasificación de paciente con Parkinson. Sin embargo, presenta algunas limitaciones como que los resultados deben ser corroborados por otras muestras independientes que confirmen los buenos resultados obtenidos. Otra posible mejora del estudio es el empleo de máquinas más potentes que permitan hacer uso de algoritmos más computacionalmente exigentes como pueden ser redes neurales o incluso el propio SVM con *kernel* polinómico.

Conclusiones

1. Los repositorios públicos son una gran fuente de datos que permiten ahorrar costes económicos y de tiempo para el planteamiento de nuevos estudios. Se puede hacer uso de múltiples *dataset* cuya combinación nos lleva a la obtención de nuevas muestras. Previo al uso de estos *dataset*, debe hacerse un estudio estadístico para conocer los datos que contienen.
2. El Machine learning es una herramienta útil para el campo de la medicina e investigación. El uso de estas herramientas es múltiple aunque siempre va a depender de la calidad de los datos de los algoritmos empleados.
3. Es posible el empleo de pruebas no invasivas para realizar predicciones en enfermedades tan complejas como la enfermedad de Parkinson. Estas pruebas pueden ser destinadas a la prevención y diagnóstico temprano de las enfermedades.

Bibliografía

1. Tysnes OB, Storstein A. Epidemiology of Parkinson's disease. 2017;124(8):901-905. doi: 10.1007/s00702-017-1686-y
2. Elbaz A. , Carcaillon L., Kab S., Moisan F. 2016. Epidemiology of Parkinson's disease. Revue Neurologique. 172(1):14-26
3. Jankovic J 2008. Parkinson's disease: clinical features and diagnosis Journal of Neurology, Neurosurgery & Psychiatry. 79:368-376
4. Little Tsanas A., McSharry M. A., Ramig L. O. 2010. Accurate Telemonitoring of Parkinsons Disease Progression by Noninvasive Speech Tests. IEEE Transactions on Biomedical Engineering. 47(4):884-893..
5. Tsanas A1, Little MA, McSharry PE, Ramig LO. 2010. Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. IEEE Trans Biomed Eng. 57(4):884-93. doi: 10.1109/TBME.2009.2036000.
6. AL-Janabi MA, Qutqut MH, Hijjawi M. 2018. Machine Learning Classification Techniques for Heart Disease Prediction: A Review. International Journal of Engeenering & Technology, 7(4):5373-5379
7. Dua, D. and Graff, C. 2019. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
8. https://www.statisticssolutions.com/missing-values-in-data/
9. https://www.theanalysisfactor.com/missing-data-mechanism/
10. Martínez González MA, Sánchez-Villegas A, Toledo Atucha EA, Faulin Fajardo J. 2014. Bioestadística amigable (3ª Edición). Elservier España. ISBN: 978-84-9022-500-4.
11. https://www.theanalysisfactor.com/mar-and-mcar-missing-data/
12. Pérez-Planells, Ll., Delegido, J., Rivera-Caicedo, J.P., Verrelst, J. 2015. Análisis de métodos de validación cruzada para la obtención robusta de parámetros biofísicos. Revista de Teledetección. 44, 55-65.
13. J. Carmona: Tutorial sobre Máquinas de Soporte Vectorial 2016. Departamento Ingeniería Artificial: Universidad Nacional de Educación a Distancia Madrid (España).
14. https://skymind.ai/wiki/random-forest
15. Little MA, McSharry PE, Roberts SJ, Costello DAE, Moroz IM. 2007. Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection, BioMedical Engineering OnLine. 6:23
16. A Tsanas, MA Little, PE McSharry, LO Ramig 2009. Accurate telemonitoring of Parkinson's

disease progression by non-invasive speech tests. IEEE Transactions on Biomedical Engineering (to appear).

17. Erdogdu Sakar, B., Isenkul, M., Sakar, C.O., Sertbas, A., Gurgen, F., Delil, S., Apaydin, H., Kursun, O. 2013. Collection and Analysis of a Parkinson Speech Dataset with Multiple Types of Sound Recordings. IEEE Journal of Biomedical and Health Informatics. 17(4):828-834

18. Sakar, C.O., Serbes, G., Gunduz, A., Tunc, H.C., Nizam, H., Sakar, B.E., Tutuncu, M., Aydin, T., Isenkul, M.E. and Apaydin, H. 2019. A comparative analysis of speech signal processing algorithms for Parkinsonâ€™s disease classification and the use of the tunable Q-factor wavelet transform. Applied Soft Computing. 74:255-263.

19. Naranjo, L., Pérez, C.J., Campos-Roca, Y., Martín, J. 2016. Addressing voice recording replications for Parkinson disease detection. Expert Systems With Applications 46: 286-292

20. Bench-Capon T., Dunne PE. 2017. Argumentation in artificial intelligence. Artificial Intelligence.171:619-641

21. Gentleman R, Huber W, Carey VJ. 2008. Supervised Machine Learning. In: Bioconductor Case Studies. 121-136

22. Le Cessie S, Van Houwelingen JC. 1992. Ridge Estimators in Logical Regression. Applied Statistics. 41(1):191-201

23. Wu X, Kumar V, Ross QJ, Ghosh J, Yang Q, Motoda H, et al. 2008. Top 10 algorithms in data minig. Knwoledge and Information Systems. 14(1):1-37.

24. Breiman L. 2001. Random forest. Machine Learning. 45(1):5-32

25. Vapnik VN. 1995. The nature of statistical learning theory. New York: Springer-Verlag

26. Goodfellow I, Bengio Y, Courville 2016. A. Deep learning. MIT Press