COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

CLASIFICACIÓN DE VARIANTES GENÉTICAS RELACIONADAS CON CARDIOPATÍAS HEREDITARIAS

Autor: Pelayo González Martínez

Director: Dido Carrero Muñiz

Madrid

Declaración de originalidad

Declaro bajo mi responsabilidad que el Proyecto presentado con el título **CLASIFICACIÓN DE VARIANTES GENÉTICAS RELACIONADAS CON CARDIOPATÍAS HEREDITARIAS** e la ETS de Ingeniería – ICAI de la Universidad Pontificia Comillas en el curso académico 4º de ingeniería de Telecomunicaciones es de mi autoría y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Uso de Inteligencia Artificial[1]

Declaro bajo mi responsabilidad que (indicar la opción correcta):

ϒ No he utilizado Inteligencia Artificial en la elaboración del presente documento.

ϒ He utilizado Inteligencia Artificial en la elaboración del presente documento y/o del Anexo B siempre en las condiciones permitidas por la Universidad Pontificia Comillas, es decir, aplicando el Nivel 2 de la Escala de Evaluación de Perkins et al. (2024): "*La IA puede utilizarse para actividades previas a la tarea, como la lluvia de ideas, la descripción y la investigación inicial. Este nivel se centra en el uso de la IA para la planificación, las síntesis y la generación de ideas, pero las evaluaciones deben hacer hincapié en la capacidad de desarrollar y refinar estas ideas de forma independiente*". En concreto, las Inteligencia Artificial ha sido empleada para:

El uso de IA se ha orientado a la búsqueda de definiciones de conceptos usados en el trabajo. Al ser un trabajo enfocado al área de la sanidad, se usa un vocabulario muy técnico del cual no tenía conocimiento al empezar a trabajar. La IA me ha permitido buscar definiciones y características de algunos conceptos, que posteriormente, con algo más de conocimiento, he verificado en otros lugares con más fiabilidad.

Firmado (alumno): Pelayo González Martínez
Fecha: 21/05/2026

Autorización para la entrega del Proyecto

El Director del Proyecto	El co-Director del Proyecto (si aplica)
CARRERO MUÑIZ DIDO - 71728888F Firmado digitalmente por CARRERO MUÑIZ DIDO - 71728888F Fecha: 2026.06.12 21:04:08 +02'00'	
Fdo:	Fdo:
Fecha:	Fecha:

[1] Esta declaración se refiere al uso de la Inteligencia Artificial generativa para realizar los documentos del Proyecto (Anexo B y Memoria). No aplica a Proyectos donde, por su naturaleza, deban emplear inteligencia artificial como parte de los mismos (aplicación de técnicas de aprendizaje automático, redes neuronales, análisis de datos…)

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

CLASIFICACIÓN DE VARIANTES GENÉTICAS RELACIONADAS CON CARDIOPATÍAS HEREDITARIAS

Autor: Pelayo González Martínez

Director: Dido Carrero Muñiz

Madrid

CLASIFICACIÓN DE VARIANTES GENÉTICAS RELACIONADAS CON CARDIOPATÍAS HEREDITARIAS

Autor: González Martínez, Pelayo.
Director: Carrero Muñiz, Dido.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Las cardiopatías hereditarias son un conjunto de enfermedades genéticas que representan un número importante de mortalidad a nivel mundial. En este proyecto se busca identificar a los pacientes que tienen más probabilidad de desarrollar una enfermedad debido a las características de sus variantes genéticas.

Palabras clave: Enfermedades, Modelos predictivos, Pacientes

1. Introducción

Las cardiopatías hereditarias son un grupo de enfermedades de origen genético que alteran directamente la estructura o el funcionamiento del corazón y que, en conjunto, suponen un número importante mortalidad en todo el mundo. Dentro de este grupo se incluyen patologías como la miocardiopatía dilatada, miocardiopatía hipertrófica o síndromes arrítmicos primarios, en los que un diagnóstico realizado con suficiente antelación puede marcar la diferencia a la hora de evitar complicaciones graves durante su desarrollo o incluso el fallecimiento de la persona.

Durante los últimos años, el desarrollo de las tecnologías de secuenciación masiva ha cambiado por completo el panorama del diagnóstico de estas enfermedades. Se han desarrollado nuevas técnicas que permiten analizar millones de fragmentos de ADN y ARN a un coste cada vez más bajo, facilitando así la identificación de un número creciente de variantes genéticas asociadas a estas enfermedades. Sin embargo, contar con más datos no implica necesariamente tener más respuestas. Identificar el impacto de todas estas variantes e identificar la o las variantes genéticas que en un individuo provocan predisposición a desarrollar una enfermedad cardiovascular sigue siendo uno de los principales retos de la genética cardiovascular actual.

ClinVar, ClinGen o LOVD son bases de datos públicas que abordan parte de estos problemas, recopilando miles de variantes genéticas aportadas por distintos laboratorios, hospitales y centros de investigación de todo el mundo, y que constituyen una fuente fundamental de evidencia a la hora de clasificarlas (Taniguti, 2022). Aun así, existen una gran cantidad de variantes que no son cubiertas por estos repositorios, y se catalogan como *variantes de significado incierto* (*VUS,* por sus siglas en ingles). Al no poder determinar si su impacto es patogénico o benigno, este tipo de variantes carecen de utilidad clínica aplicable, lo que evidencia aún más la necesidad de desarrollar herramientas analíticas más sólidas y consistentes.

En este contexto, las técnicas de *Machine Learning* se perfilan como una alternativa especialmente prometedora. Esto se debe a su gran capacidad para detectar patrones los grandes volúmenes de datos y su fácil adaptabilidad a diferentes ámbitos. Esta línea es

la que busca seguir el trabajo, siendo capaz al final del todo de discriminar entre variantes patogénicas y benignas asociadas a cardiopatías hereditarias.

2. Definición del proyecto

El trabajo tiene como finalidad principal el diseño, desarrollo y validación de un modelo de aprendizaje automático que sea capaz de clasificar variantes genéticas relacionadas con cardiopatías hereditarias, discriminando entre aquellas que son patogénicas y benignas.

Antes de empezar a trabajar en el modelo, se deben recopilar y preparar los datos que vamos a usar para elaborarlo. Estos datos se han extraído de bases de datos públicas, y se han aplicado posteriormente técnicas de limpieza atendiendo a definiciones y fórmulas matemáticas.

Una vez realizada la limpieza del dataset, se empieza a elaborar el modelo donde se implantan diferentes algoritmos, ajustando parámetros hasta encontrar el que más se ajusta a nuestro objetivo. El rendimiento de estos se evaluará usando métricas tales como la precisión y la precisión balanceada.

Por último, con el fin de evaluar la consistencia de los modelos, se realizará una validación externa con distintos datasets. Una vez realizada esta validación, podremos concluir si el modelo permite la identificación certera de variantes patogénicas en el contexto de la enfermedad cardiovascular.

3. Descripción del modelo/sistema/herramienta

El modelo propuesto para el desarrollo del trabajo está basado en XGBoost. Este algoritmo se basa en árboles de decisión secuenciales donde cada uno va corrigiendo los errores de los anteriores. XGBoost ofrece una alta velocidad y escalabilidad, junto con el manejo de valores nulos, muy útil para tablas con variables sin valor en varios datos, como ocurre con nuestro dataset.

El modelo se ha desarrollado completamente usando Python en JupyterLab debido a la gran cantidad de librerías que nos brindan opciones de modelos y representaciones de variables.

El sistema toma como entrada una base de datos de gran tamaño (59377 ejemplos), y la procesa para eliminar posibles atributos prescindibles. Posteriormente, haciendo uso de librerías, se ajustan los parámetros del algoritmo y se entrena el mejor modelo. Con este se realiza una validación externa con otros datasets y se extraen conclusiones.

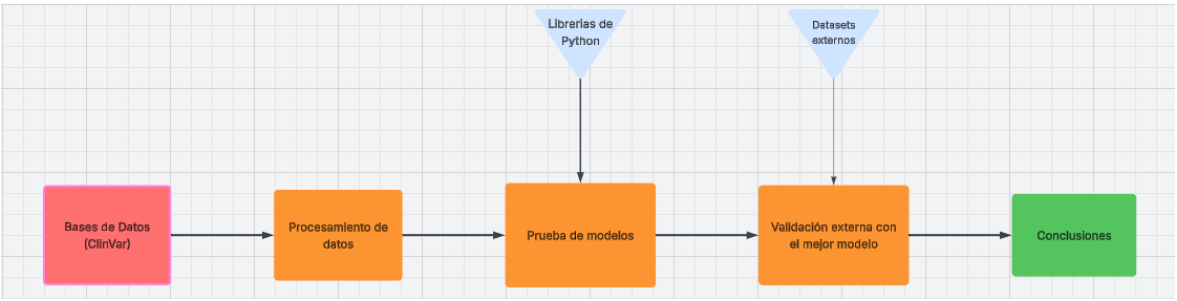

Figura 1 - Diagrama de flujo del sistema planteado

4. Resultados

Durante el preprocesamiento se observa que existen variables que no contienen valores en más de 100 datos (0.1% del total de datos). Estas variables se eliminan y se realiza un primer modelo, modelo base con el que podremos comparar. Al comenzar el trabajo, se ha realizado un análisis de correlación entre atributos para determinar cuáles son prescindibles.

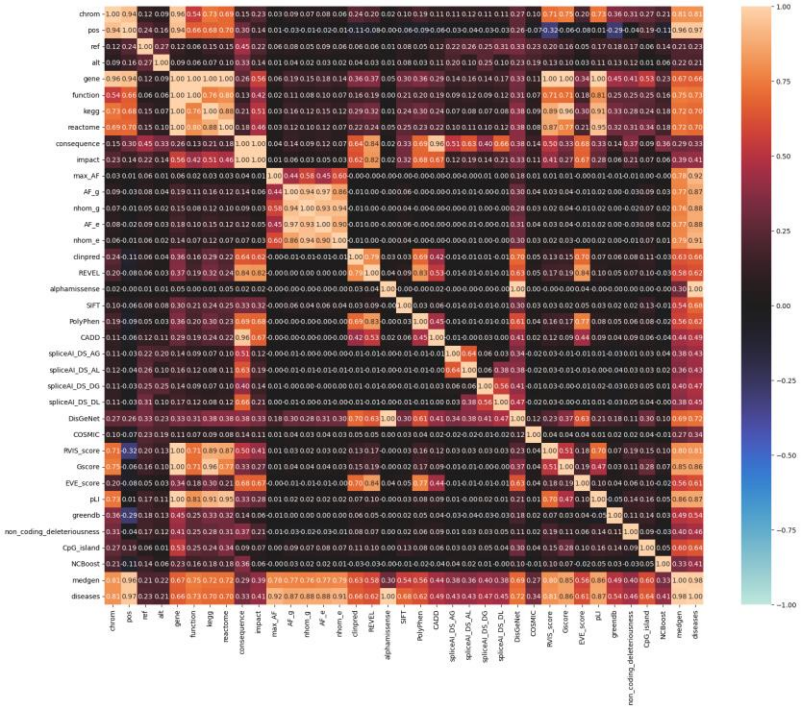

Figura 2 - Correlación entre atributos del dataset

Se han seguido dos líneas de trabajo. La rama que mejores resultados ha demostrado ha sido en la que se ha realizado un filtro de atributos basándose en su definición. Para ir mejorando el modelo en esa rama, se ha usado el análisis SHAP, que nos muestra la importancia de determinadas variables a la hora de determinar la variable dependiente.

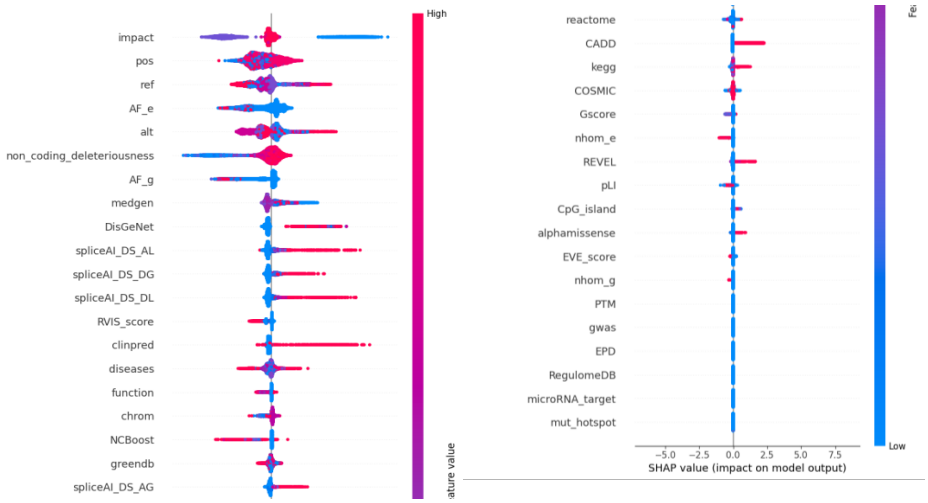

Figura 3 - Análisis SHAP de la segunda rama

Estableciendo el límite de importancia a distintos niveles, el mejor modelo ha resultado de poner el límite de importancia en la variable *Revel*, con el que obtenemos una precisión del 98.88% y una precisión balanceada de 97.57%.

Para certificar la verdadera robustez de mi modelo predictivo y descartar cualquier tipo de sobreajuste, llevé a cabo un proceso de validación externa con el objetivo de simular su despliegue en un escenario clínico real frente a datos completamente nuevos. Inicialmente seleccioné la base de datos ClinGen, pero los resultados arrojaron una precisión perfecta irreal del 100% que evidenció un problema en los datos. El algoritmo no estaba generalizando, sino recordando variantes que ya se encontraban en mi conjunto de entrenamiento original. Ante este escenario fallido, decidí evaluar el modelo con una segunda alternativa, SpadaHC, una base de datos española especializada en cáncer hereditario que introdujo un reto mayor al someter al sistema a una nueva variabilidad clínica y poblacional. Aunque me vi en la obligación de eliminar dos variables incompatibles para poder operar, esta validación fue un éxito rotundo. El modelo alcanzó un 96.23% de precisión y un Área Bajo la Curva de 0.9934, demostrando matemáticamente que había aprendido las reglas biológicas de la patogenicidad y sabía separar casi a la perfección las variantes inofensivas de las nocivas. Finalmente, para compensar la información perdida tras la eliminación de las variables y pulir el rendimiento general, decidí buscar una manera de mejorar el modelo a muy bajo coste para estos datos, un ajuste fino que logró elevar la precisión definitiva hasta un excelente 97.10%.

5. Conclusiones

Los atributos que más peso tienen en el mejor modelo obtenido son *Impact, pos, ref, alt* y *AF_e* entre otros. Estos hacen referencia al impacto de la variante predicho por VEP, coordenadas cromosómicas en las que se encuentra la variante, nucleótido de la referencia y nucleótido al que se produce el cambio y frecuencia poblacional de la variante, según gnomAD v4 de exomas respectivamente.

Por otro lado, valores de mut_hotspot, microRNA_target o EPD, no tienen gran relevancia en el modelo. Estos hacen referencia a presencia (1) o ausencia (vacío) de la variante en bases de datos de hotspots mutacionales, presencia (1) o ausencia (vacío) de la variante en bases de datos de sitios de unión a microRNA y referencia a la región promotora en la que se sitúa la variante o campo vacío si no hay información según EPD https://epd.expasy.org/epd/EPDnew_select.php respectivamente.

Además, el proyecto se caracteriza por su rigurosidad metodológica y por la auditoría detallada de datos. La etapa de validación externa demostró la madurez analítica del desarrollo al identificar la fuga de información que sucedía al entrecruzar las bases de datos de ClinVar y ClinGen.

Finalmente, lo que más sobresale del proyecto es la evidencia empírica de la universalidad y validación transversal del modelo. Cuando se llevó el algoritmo, originalmente de la genética cardiovascular, al ámbito oncológico enfocado en el cáncer hereditario en la cohorte SpadaHC, el sistema continuó con métricas de rendimiento excepcionales. El AUC alcanzó el 0.9934 y la exactitud balanceada llegó al 97.21% tras buscar un modelo óptimo para esos datos. Estos resultados sobresalientes demuestran de forma indiscutible que el algoritmo ha conseguido extraer con éxito las reglas biológicas universales de la patogenicidad, lo que fortalece mi desarrollo como una herramienta de soporte diagnóstico sólida y totalmente agnóstica a la enfermedad específica bajo evaluación.

6. Referencias

Clinical Genome. (s.f.). *Clinicalgenome*. Obtenido de clinicalgenome.org: https://www.clinicalgenome.org/genomeconnect/for-patients-genomeconnect/old-faq/preguntas-frecuentes-spanish-faq/

Grifol, D. (27 de Febrero de 2026). *DanielGrifol*. Obtenido de danielgrifol.es: https://danielgrifol.es/curva-de-productividad-diaria/

Lundberg, S. M. (2017). *A Unified Approach to Interpreting Model Predictions*. Obtenido de papers: https://papers.nips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Minton, K. (2023). *Predicting variant pathogenicity with Alphamissense.* Nature Reviews Genetics.

CLASSIFICATION OF GENETIC VARIANTS RELATED TO INHERITED HEART DISEASE

Author: González Martínez, Pelayo.
Supervisor: Carrero Muñiz, Dido.
Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

Keywords: Diseases, Predictive models, Patients

1. Introduction

Inherited heart diseases are a group of genetic disorders that directly affect the structure or function of the heart and, collectively, account for a significant number of deaths worldwide. This group includes conditions such as dilated cardiomyopathy, hypertrophic cardiomyopathy, and primary arrhythmic syndromes, in which an early diagnosis can make all the difference in preventing serious complications as the disease progresses or even death.

In recent years, the development of next-generation sequencing technologies has completely transformed the landscape. New techniques have been developed that allow for the analysis of millions of DNA and RNA fragments at an increasingly lower cost, thereby facilitating the identification of a growing number of genetic variants associated with these diseases. However, having more data does not necessarily mean having more answers. Identifying the impact of all these variants remains one of the main challenges in current cardiovascular genetics.

ClinVar, ClinGen, and LOVD are public databases that address some of these issues by compiling thousands of genetic variants contributed by various laboratories, hospitals, and research centers around the world, and they serve as a key source of evidence for classifying these variants (Taniguti, 2022). Even so, there are a large number of variants not covered by these repositories, which are classified as variants of uncertain significance (VUS). Since it is impossible to determine whether their impact is pathogenic or benign, these variants lack clinical utility, further highlighting the need to develop more robust and consistent analytical tools.

Given the context described above, machine learning techniques are emerging as a particularly promising alternative. This is due to their remarkable ability to detect patterns in large volumes of data and their ease of adaptation to different fields. This is the approach this study aims to follow, with the goal of distinguishing between pathogenic and benign variants associated with hereditary heart diseases.

2. Definition of the proyect

The main aim of this project is to design, develop and validate a machine learning model capable of classifying genetic variants associated with inherited heart conditions, distinguishing between those that are pathogenic and those that are benign.

Before beginning work on the model, the data to be used must be collected and prepared. This data has been extracted from public databases, and data cleaning techniques have subsequently been applied in accordance with mathematical definitions and formulas.

Once the dataset has been cleaned, work begins on building the model, implementing various algorithms and adjusting parameters until the one that best meets our objective is found. The performance of these models will be evaluated using metrics such as accuracy and balanced accuracy.

Finally, in order to assess the consistency of the models, external validation will be carried out using different datasets. Once this validation has been completed, we can begin to draw conclusions and identify limitations…

3. Description of the model

The model proposed for this project is based on XGBoost. It utilises sequential decision trees, with each subsequent tree correcting the errors of the previous ones. XGBoost offers high speed and scalability, along with effective handling of missing values, which is particularly useful for tables containing missing attribute values in various datasets.

The model has been developed entirely using Python in JupyterLab due to the large number of libraries that provide us with options for models and variable representations.

The system takes a massive database as input and processes it to eliminate any potentially redundant attributes. Subsequently, using libraries, the parameters are adjusted and the best model is created. This model is then used to perform external validation with other datasets, and conclusions are drawn.

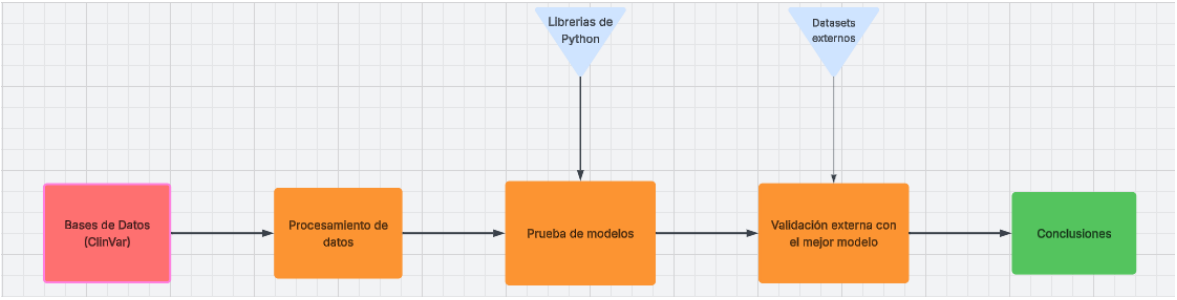

Figura 4 - Flowchart of the system planned

4. Results

Upon initially visualising the data, we can see that there are attributes which do not contain values in more than 100 instances. These are immediately removed, and an initial model baseline model—is created, which we can then use for comparison. At the start of the project, a correlation analysis was carried out between attributes to determine which ones could be omitted.

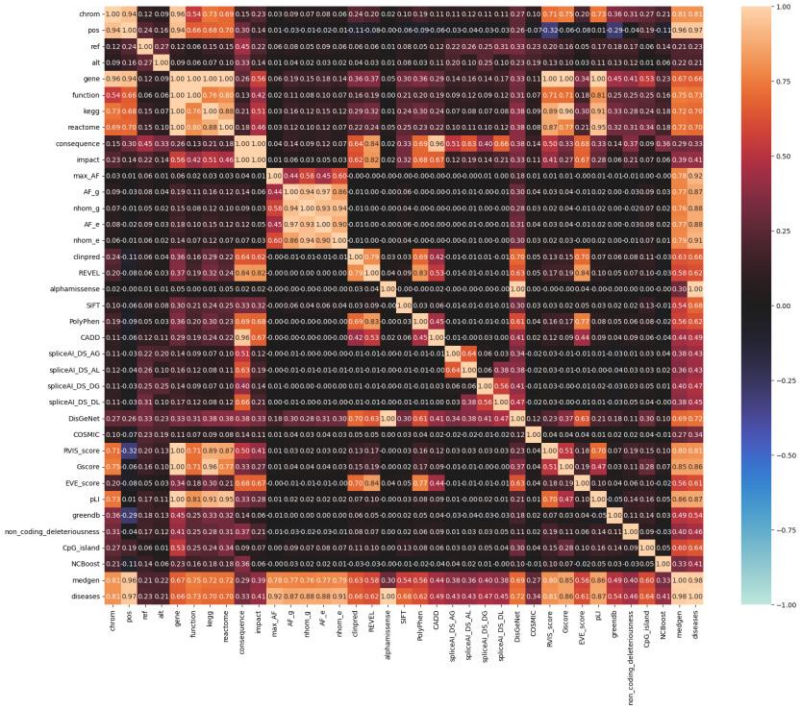

Figura 5 - Correlation between dataset attributes.

Two lines of work have been pursued. In one, an initial attribute filtering process was carried out, based on the definition of the attributes, whereby variables were removed that were considered, by virtue of their definition, to be covered by others.

The approach that has yielded the best results is the one in which an attribute filter was applied based on their definition. To further improve the model in this approach, Shap analysis was used, which shows the importance of certain variables in determining the dependent variable.

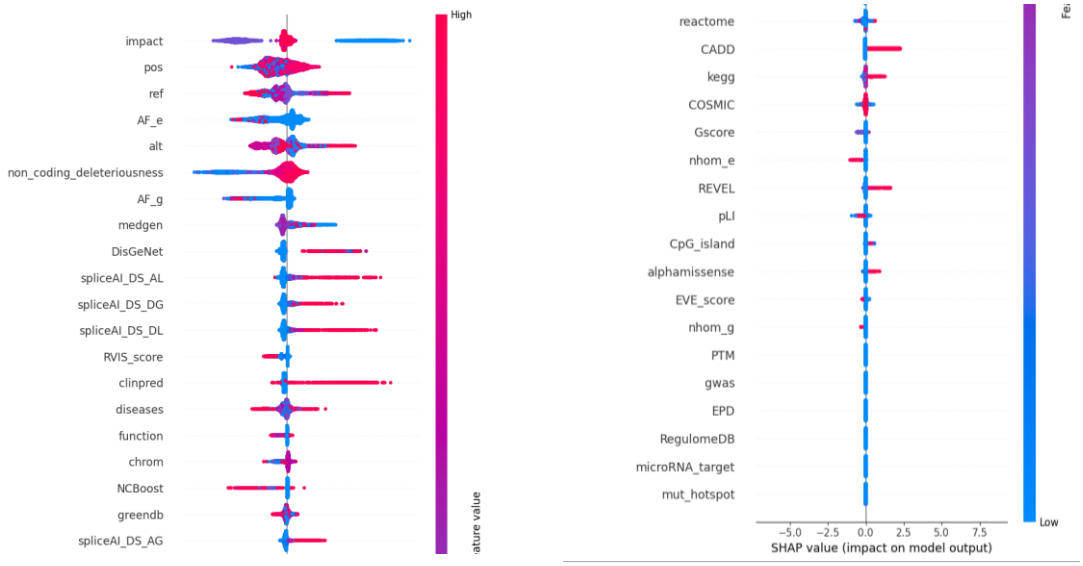

Figura 6 - SHAP analysis of the second line of work.

By setting the materiality threshold at different levels, the best model was obtained by setting the threshold at Revel, where we achieved an accuracy of 98.88% and a balanced accuracy of 97.57%.

To certify the true robustness of my predictive model and rule out any type of overfitting, I carried out an external validation process aimed at simulating its deployment in a real clinical scenario against completely new data. Initially, I selected the ClinGen database, but the results yielded an unrealistically perfect accuracy of 100%, which highlighted a data leakage problem. The algorithm was not generalizing, but rather memorizing variants that were already present in my original training set. Faced with this setback, I decided to evaluate the model using a second alternative: SpadaHC, a Spanish database specialized in hereditary cancer. This introduced a greater challenge by subjecting the system to new clinical and population variability. Although I was forced to remove two incompatible variables in order to proceed, this validation was a resounding success. The model achieved an accuracy of 96.23% and an Area Under the Curve of 0.9934, mathematically demonstrating that it had learned the biological rules of pathogenicity and could separate harmless variants from harmful ones almost perfectly. Finally, to compensate for the information lost after the removal of the variables and to polish the overall performance, I sought a way to improve the model at a very low computational cost for this data; a fine-tuning process that managed to raise the final accuracy to an excellent 97.10%.

5. Conclusions

The attributes that carry the most weight in the best model obtained are Impact, pos, ref, alt and AF_e, amongst others. These refer to the impact of the variant as predicted by VEP, the chromosomal coordinates where the variant is located, the reference nucleotide and the nucleotide at which the change occurs, and the population frequency of the variant, according to gnomAD v4 exome data respectively.

On the other hand, values for mut_hotspot, microRNA_target or EPD are not particularly significant in the model. However, they are not dispensable. These refer to

the presence (1) or absence (empty) of the variant in mutational hotspot databases, the presence (1) or absence (empty) of the variant in microRNA binding site databases, and a reference to the promoter region in which the variant is located or an empty field if no information is available, according to EPD https://epd.expasy.org/epd/EPDnew_select.php respectively.

Furthermore, the project is characterized by its methodological rigor and detailed data auditing. The external validation stage demonstrated the analytical maturity of the development by identifying the data leakage that occurred when cross-referencing the ClinVar and ClinGen databases.

Finally, what stands out most about the project is the empirical evidence of the model's universality and cross-domain validation. When the algorithm, originally from cardiovascular genetics, was applied to the oncology domain focusing on hereditary cancer in the SpadaHC cohort, the system maintained exceptional performance metrics. The AUC reached 0.9934 and the balanced accuracy hit 97.21% after searching for an optimal model for that data. These outstanding results undeniably demonstrate that the algorithm has successfully extracted the universal biological rules of pathogenicity, which solidifies my development as a robust diagnostic support tool that is completely agnostic to the specific disease under evaluation.

6. References

https://www.clinicalgenome.org/genomeconnect/for-patients-genomeconnect/old-faq/preguntas-frecuentes-spanish-faq/

Grifol, D. (27 de Febrero de 2026). *DanielGrifol*. Obtenido de danielgrifol.es: https://danielgrifol.es/curva-de-productividad-diaria/

Lundberg, S. M. (2017). *A Unified Approach to Interpreting Model Predictions*. Obtenido de papers: https://papers.nips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Minton, K. (2023). *Predicting variant pathogenicity with Alphamissense.* Nature Reviews Genetics.

.

Índice de la memoria

Índice de figuras

Índice de tablas

Capítulo 1. INTRODUCCIÓN

El corazón es el motor de nuestro cuerpo. Late unas cien mil veces al día de forma automática, bombeando vida a nuestras extremidades en un ciclo incesante que no requiere nuestra atención. Al ser el órgano fundamental para nuestro funcionamiento, merece ser estudiado y tratado con detenimiento para evitar contratiempos. Mucha gente da por sentado que, si se cuidan, el corazón no tendrá ningún problema. No obstante, ¿qué ocurre cuando el diseño original de nuestro motor alberga alguna vulnerabilidad invisible desde el instante en el que nacemos?

Este tipo de errores son las llamadas cardiopatías hereditarias. A diferencia de otras afecciones cardiovasculares que pueden resultar de un mal estilo de vida, una dieta poco equilibrada o del propio envejecimiento del cuerpo, estas patologías tienen un origen anterior. Constituyen un conjunto de enfermedades genéticas que afectan tanto a la estructura física como a la capacidad funcional del corazón y, hoy en día, representan una causa de morbilidad y mortalidad con un gran volumen de pacientes a nivel mundial. Unos de los motivos es la naturaleza de las propias enfermedades, ya que suelen ser silenciosas y ocultarse en individuos que aparentar estar sanos.

El peso de estas enfermedades recae intensamente en el concepto de la herencia, al estar "condenado" a esta y no poder evitarla, en principio. Ahí es donde, en parte, reside la cuestión de este trabajo. La genética que habitualmente asociamos, de forma romántica, a heredar el color de los ojos de un abuelo o la nariz de la madre, se transforma en nuestro contexto médico en una advertencia. Detrás de un paciente hay unas raíces que explican las características de este, y muchas veces son causa de incertidumbre y preocupación a su alrededor.

1.1 CONCEPTOS

Para garantizar que el lector comprende correctamente el trabajo, se van a definir previamente una serie de términos técnicos que pueden resultar desconocidos para las personas que no estén familiarizadas con el sector.

Un concepto que va a estar resonando constantemente a lo largo del proyecto es el de cardiopatía hereditaria. Las cardiopatías hereditarias (CH) son un grupo de enfermedades cardiovasculares con una base genética, con presentación familiar y asociadas en gran parte a la muerte súbita. Dentro de estas se incluyen miocardiopatías, canalopatías, enfermedades aórticas familiares… (Reyes, 2022)

Otro concepto que debemos conocer para contextualizar nuestro trabajo es la definición de Secuenciación Masiva o NGS (Next Generation Sequencing). Estas son tecnologías de laboratorio vanguardistas que nos permiten realizar el análisis de millones de fragmentos de ADN y ARN de manera simultánea. Estas nos dotan con bases de datos enormes sobre las que, realizando las operaciones necesarias, se pueden sacar conclusiones muy relevantes.

Por último, otro concepto fundamental en el trabajo. Y, en parte, una de las motivaciones por realizar este, son las VUS (Variantes de significado incierto). Estas son alteraciones presentes en el ADN que, al no contar con evidencia suficiente, no se conoce su efecto. Al no poder clasificar estas variantes, los médicos requieren de modelos robustos que sean capaces de analizar datos y predecir su efecto.

1.2 MOTIVACIÓN DEL PROYECTO

La realización de este proyecto viene motivada por la necesidad inminente de mejorar el diagnostico de las cardiopatías, atendiendo a las limitaciones que estas presentan actualmente en el ámbito clínico. SI no, el paciente queda en un limbo en el que no se puede beneficiar de determinadas terapias o ensayos clínicos al no identificar la causa de su enfermedad. A pesar de la inmensa cantidad de datos disponibles hoy en día, una proporción considerable de variantes presentes en estas bases gigantescas permanecen clasificadas como

inciertas. La gran cantidad de variantes genéticas, pacientes, síntomas, etc, no se trata de un mero problema de almacenamiento de datos, presentan una barrera que dificulta la toma de decisiones clínica y reducen el impacto de la secuenciación genómica.

"El tipo de herencia y los genes involucrados varían en las diferentes ECV. La mayoría de las CH son enfermedades monogénicas, donde una sola variante genética es suficiente para producir la enfermedad. En estos casos… existe una probabilidad del 50 % de transmitir la variante a cada descendiente." (Reyes, 2022). Frente a esta situación, se busca desarrollar un modelo de Machine Learning que ayuda a clasificar variantes como patogénicas o benignas, para atajar, con suficiente anterioridad, una posible enfermedad. No se pretende sustituir el juicio clínico, sino aportar una ayuda apoyando a los especialistas con herramientas robustas y escalables que sean capaz de trabajar y evolucionar a medida que los datos crecen. De esta manera, mejoramos el asesoramiento genético, que consiste en el proceso de explicar las consecuencias de la enfermedad, las probabilidades de padecer los síntomas y los posibles tratamientos junto con el impacto que puede tener en la descendencia. (Resta , et al., 2006)

.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

En este capítulo se van a presentar las herramientas, librerías, modelos… que se han usado para lograr el objetivo de este proyecto. Se profundizará en las tecnologías que han tenido más impacto dentro del trabajo.

2.1 PLATAFORMAS DE DATOS

Cualquier modelo de aprendizaje automático realizado correctamente, está respaldado por grandes bases de datos desde donde se saca información para el análisis. En este caso, la información utilizada no forma parte de bases de datos relacionales convencionales, sino que se apoya en repositorios internacionales como **ClinVar** y **ClinGen**. ClinVar es una base de datos publica mantenida por el NCBI donde todos los investigadores suben las clasificaciones de sus variantes una vez han hecho experimentos para evaluar si son patogénicas o benignas. Debido a la dificultad de encontrar datasets independientes con clasificaciones y cuyas variantes no las hayan subido a ClinVar, se ha usado como la fuente principal de datos del proyecto. ClinGen por otro lado es una iniciativa respaldada por el NIH centrada en entender la función de las alteraciones genéticas en la salud de las personas. Este es el encargado de la recolección, almacenamiento y utilización compartida de datos. (Clinical Genome, n.d.) Tanto ClinGen como **SpadaHC** se han utilizado como fuentes de validación externa. SpadaHC es una base de datos centrada en recopilar mutaciones que se relacionan con cáncer hereditario en España.

2.2 LENGUAJE Y ENTORNO DE DESARROLLO

Python es el lenguaje de programación que se ha utilizado para diseñar estos modelos. ¿Por qué Python y no otros? Python proporciona un ecosistema muy completo de paquetes que nos permite explorar el problema desde varios ángulos. JupyterLab nos ayuda a organizar los modelos por carpetas para separar los modelos y mantener el entorno bien ordenado.

Las librerías que se han usado en el trabajo son: `pandas`, `Numpy`, `Scikit-Learn`, `XGBoost`, `Seaborn`, `Matplotlib` y `SHAP`.

Pandas es la herramienta que nos permite leer los datos, y manipular los dataframes. Nos ayuda a tener una visión más clara de los datos (valores nulos, tipo del atributo…), sus características (tamaño del dataset) y a realizar una correcta limpieza de datos. Numpy por otro lado, traduce estas tablas en matrices permitiéndonos realizar operaciones con estas. Gracias a estas estructuras podemos encontrar relaciones entre los datos y usarla como herramienta de limpieza de datos.

Scikit-Learn es la biblioteca principal para Machine Learning en el entorno de Python. Gracias a esta podemos entrenar, modelar y evaluar los modelos de aprendizaje automático desarrollados. Gracias a esta somos capaces de saber el rendimiento de los modelos en base a métricas como la precisión. Esta librería trabaja mano a mano con XGBoost, librería que nos aporta el propio modelo de aprendizaje que vamos a usar.

Tanto Matplotlib como Seaborn y SHAP son librerías que nos facilitan un análisis más interpretativo de los datos mediante representaciones visuales. Matplotlib trabaja con los datos facilitados y Seaborn se encarga de representarlos. SHAP se trata de una librería más compleja ya que incorpora ambas funciones y nos permite saber el orden de importancia dentro de los modelos de los atributos.

Capítulo 3. ESTADO DE LA CUESTIÓN

El sector de la Investigación Académica, la intersección entre la inteligencia artificial y las miocardiopatías es un área de investigación muy activa. Esto es debido a la complejidad de la arquitectura genética.

El consorcio europeo SMASH-HCM y otros proyectos están explorando la posibilidad de combinar datos genéticos, imágenes médicas como resonancias magnéticas e inteligencia artificial para desarrollar "gemelos digitales" que pronostiquen el riesgo y el desarrollo de los pacientes. En el ámbito especializado de la genómica de secuencias, en el que se encuentra este proyecto, una parte importante de la bibliografía más reciente se enfoca en capacitar algoritmos clásicos supervisados como Máquinas de Vectores de Soporte, Random Forest o la Regresión Logística sobre genes muy específicos y críticos, por ejemplo, el gen MYBPC3, uno de los mayores responsables de la miocardiopatía hipertrófica.

En los últimos años, debido al auge del desarrollo de técnicas de aprendizaje automático, han surgido diferentes plataformas computacionales con el objetivo de asistir a los profesionales. Dentro de estos programas, muchos se han enfocado en el desarrollo de herramientas específicas para la cardiología.

Uno de los más conocidos es AlphaMissense, desarrollado por Google DeepMind. Se trata de una inteligencia artificial que busca predecir la patogenicidad de variantes en la secuencia del ADN, adaptado de su herramienta de predicción de estructura de proteínas AlphaFold. (Minton, 2023) Hasta el momento, solo el 0.1% de las mutaciones que parecían no tener sentido, habían sido clasificadas por expertos humanos. AlphaMissense surge como una alternativa revolucionaria al haber logrado clasificar el 89% de estas variaciones genéticas. (Parra.S, Aphamissense: la IA de Google que revoluciona la detección de mutaciones geneticas. , 2023, September 25)

A nivel más específico, las plataformas más destacadas son CardioClassifier y CardioBoost. CardioClassifier es una herramienta web automatizada e interactiva que facilita la interpretación específica de variantes genéticas en genes asociados con afecciones cardíacas hereditarias. CardioBoost por otro lado, es una herramienta parecida en cierta parte a la nuestra. CardioBoost es un clasificador de variantes para enfermedades. Este se caracteriza por una alta precisión de discriminación global. Incluye dos clasificadores de variantes particulares para dos conjuntos de síndromes que están íntimamente vinculados: uno para la cardiomiopatía dilatada y la hipertrófica, que forman parte de las cardiomiopatías familiares, y otro para los síndromes de arritmia hereditarios, que comprenden el síndrome de Brugada y el síndrome de QT largo.

Este proyecto en este entorno de innovación avanzada no tiene como objetivo competir a nivel comercial con arquitecturas masivas como las de DeepMind. Más bien, pretende agregar valor a través de la transparencia y la interpretabilidad. El propósito es proporcionar una metodología que sea explícita, replicable y que un genetista clínico pueda auditar con facilidad, empleando datos públicos y centrándose exclusivamente en las variables de mayor impacto en el fenotipo cardiovascular.

Creo que te envie un qrticulo recientemente, no? Asegurate de incluirlo; se publico hace un mes y eso va a demostrar que estas pendiente de la literaatura reciente en tu materia

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Si la motivación expuesta en la introducción de este proyecto respondía a una vocación con impacto social y humano, la necesidad ética de reducir la mortalidad y arrojar luz sobre la incertidumbre de los pacientes, la justificación funcional de este trabajo obedece a una demanda estricta, tangible y urgente del mercado tecnológico y clínico actual.

A la vista del análisis realizado en el estado de la cuestión, resulta innegable que la medicina de precisión y la genómica cardiovascular han superado un hito histórico. Hoy en día, gracias a la secuenciación de nueva generación (NGS) hemos adquirido la capacidad de secuenciar el panel genético de un número impensable de pacientes y convertirlo en un proceso rutinario y económicamente accesible. Pero, como dice el Ying y el Yang, todo tiene luces y sombras. Este triunfo tecnológico ha chocado frontalmente contra un llamado *cuello de botella*, la interpretación de la información. El primer paso de abaratar la capacidad de leer el genoma ya está completado, sin embargo, descifrar que implican exactamente esas alteraciones sigue siendo un proceso lento y no del todo pulido en el ecosistema de la salud.

Precisamente este escenario de ineficiencia operativa es donde se sitúa nuestro trabajo y le da su por qué. Actualmente tenemos un claro vacío en el sector de la salud digital. Muchas decisiones tomadas en el día a día por los médicos se toman sin ser conocedores al 100% del estado del paciente. Y esto no siempre es culpa suya ya que hay procesos que se demoran más y a veces se les exigen tomar medidas inmediatas para evitar posibles problemas más graves. Por un lado, se ven forzados a conservar procesos de validación lentos y casi artesanales, que dependen de personal con alta especialización que tiene que buscar manualmente en bases de datos inconexas. Por otro lado, la aparición de las grandes empresas tecnológicas ha saturado el mercado con macro modelos predictivos que, aunque son prodigiosos en términos matemáticos, funcionan como "cajas negras" oscuras y

generalistas que se desarrollan sin considerar la necesidad imperante de auditar las decisiones en el ámbito cardiovascular.

Este trabajo busca crear una herramienta estratégica que tanto un director de laboratorio o un inversor en tecnología médica desearía implementar inmediatamente en su cadena de valor. Uno de los valores diferenciales que hacen de este proyecto una propuesta robusta y necesaria es la capacidad directa de transformar una avalancha de datos genómicos no estructurados en conocimiento clínico accionable y rentable. Para articular de forma precisa esta propuesta de valor, la viabilidad e idoneidad del proyecto se fundamenta y desglosa a continuación a través de tres pilares fundamentales: la justificación técnica, el entorno clínico-económico y las sinergias institucionales.

4.1.1 JUSTIFICACIÓN TÉCNICA: INTERPRETABILIDAD Y ESPECIALIZACIÓN

Desde una perspectiva puramente técnica, el mercado actual del diagnóstico genómico está polarizado y dominado por herramientas que no terminan de encajar en el flujo de trabajo del genetista cardiovascular.

Por un lado, la industria ha desarrollado modelos predictivos de carácter generalista. Destacan soluciones impulsadas por grandes corporaciones biotecnológicas como Illumina, que han sido capaces de desplegar arquitecturas de aprendizaje profundo como PrimateAI, o herramientas de predicción de empalme genético como SpliceAI. Estos modelos, pese a estar entrenados sobre grandes volúmenes de datos de ClinVar proporcionándoles una capacidad predictiva fiable, presentan una debilidad fundamental para el profesional encargado de diagnosticar al paciente. La gran mayoría se centran en evaluar variantes de forma genérica a lo largo de todo el exoma humano, en lugar de estar optimizados para las particularidades y los patrones clínicos específicos de las cardiopatías hereditarias.

Por otro lado, la irrupción de modelos de Deep Learning, como AlphaMissense de Google DeepMind comentado anteriormente, ha supuesto un hito en la predicción de patogenicidad, pero se tratan de modelos de “caja negra”. En el ámbito médico, un cardiólogo no puede ni ética ni legalmente, basar una decisión que pueda poner en riesgo al cliente, como la

implantación preventiva de un desfibrilador en un paciente joven, en el resultado de una red neuronal la cual no son conocidos los procesos internos que se atraviesan a la hora de tomar una decisión.

Para hacer frente a estas soluciones opacas, este trabajo propone el desarrollo de modelos de Machine Learning supervisados, transparentes y centrados en los fenotipos cardiovasculares. Gracias a ser capaz de acotar el dominio del problema y emplear algoritmos cuya importancia de características, *feature importance*, puede ser aislada y auditada, se ofrece al mercado como una herramienta usada para asistir a un verdadero experto. No se busca sustituir al médico como tal, sino que le proporciona una predicción respaldada por variables clínicas donde se exponen las variables que más peso han tenido en la decisión, aportando confianza en la estructura general del modelo.

4.1.2 JUSTIFICACIÓN CLÍNICA Y DE MERCADO: VIABILIDAD Y EFICIENCIA

Desde el punto de vista de la gestión hospitalaria y el incipiente mercado de la salud digital (*e-health*), el tiempo y el talento humano son, con diferencia, los recursos más costosos y tensionados.

En la práctica clínica habitual, herramientas de apoyo como CardioClassifier ayudan a estandarizar las reglas del protocolo ACMG (*American College of Medical Genetics and Genomics)*, pero siguen exigiendo una constante curación y revisión manual. A veces, la secuenciación de un panel genético devuelve una Variante de Significado Incierto (VUS), ese paciente entra en lo que podemos denominar, una “odisea diagnostica” por la dificultad de clasificar esas variantes. Cuando se da esa situación, el laboratorio de genética se ve abocado a un arduo proceso de investigación artesanal. Una revisión manual de literatura científica que consiste en consultar repositorios aislados, estudios de segregación familiar y costosos ensayos. Este flujo de trabajo es muy lento y puede demorarse semanas e incluso

meses, consumiendo decenas de horas de personal altamente cualificado para cada variante descubierta.

La implementación en el mercado de estos modelos que son capaces de automatizar y escalar fácilmente la criba inicial, ofreciendo a la sanidad pública o laboratorios privados un ROI (*Return of Investment*) positivo gracias a:

- Una optimización de recursos humanos: Es capaz de convertir un proceso de revisión documental heurística de días en una inferencia computacional capaz de ejecutarse en milisegundos. Esto permite a bioinformáticos y genetistas liberar su agenda y ganar mucho más tiempo para centrarse exclusivamente en las variantes verdaderamente complejas o novedosas, delegando la clasificación preliminar de las mutaciones en el algoritmo.

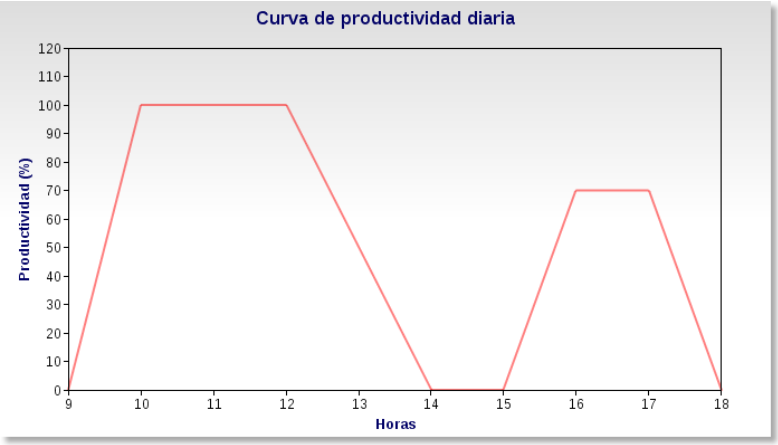

Figura 7 - Curva de productividad diaria (Grifol, 2026)

Según Daniel Grifol, una persona que ha probado todos los sistemas de productividad que existen y basándose en un estudio de Harvard Business School y BCG que concluyen que los profesionales que usan IA completan tareas un 25% más rápido y con un 40% más de calidad, ha determinado el grafico anterior de la productividad diaria en un día de trabajo. Nuestro trabajo busca aportar una herramienta a los trabajadores de los laboratorios o las clínicas para que no malgasten el tiempo de trabajo en tareas que se pueden realizar automáticamente, y así sacar un mayor rendimiento de los trabajadores y, por tanto, de la empresa.

- Aceleración en la toma de decisiones clínicas y ahorro hospitalario: Una clasificación rápida y precisa de patogenicidad permite cerrar el diagnostico definitivo del paciente de forma temprana. Esto mejora la administración completa de los servicios cardiológicos, disminuye la prescripción de pruebas diagnósticas repetitivas que son costosas y previene ingresos hospitalarios en el futuro al permitir que se prescriban tratamientos preventivos, por ejemplo, desfibriladores implantables o betabloqueantes, antes de una crisis cardíaca. Un diagnóstico temprano en un paciente puede detectar una enfermedad invisible a priori y en los mejores casos, puede resultar en salvar una vida.
- Reducción de la carga psicológica y mejora del servicio: Tanto a nivel humano como comercial, la capacidad de ofrecer a los pacientes y a sus familias una respuesta clínica concluyente en un periodo de tiempo reducido mejora considerablemente la calidad del servicio de salud y disminuye la grave ansiedad asociada a los historiales familiares de muerte súbita.

4.1.3 JUSTIFICACIÓN EVOLUTIVA Y VALOR INTERDISCIPLINAR

Las guías de clasificación en papel o los sistemas informáticos estáticos, requieren mucho tiempo y consenso internacional para ser actualizados o modificados. En este mercado tan vanguardista como la genómica cardiovascular, donde el volumen de descubrimientos de nuevas mutaciones crece exponencialmente cada mes, por ende, depender de herramientas estáticas provoca que los laboratorios trabajen con metodologías que corren el riesgo de quedar obsoletas con rapidez.

Frente a esta rigidez metodológica surge el desarrollo de este trabajo aportándole la dinámica intrínseca en machine learning. Los modelos creados no son modelos de clasificación cerrados. Funcionan como sistemas evolutivos con la capacidad de ir aprendiendo continuamente, y transparente con sus operaciones. A medida que las bases de datos van incrementando, el modelo se va alimentando de estas y puliendo su toma de decisiones ajustando los umbrales de decisión lo que le permite mejorar su precisión sin reescribir su código base ni rediseñar su arquitectura desde cero.

Por último, este trabajo consigue integrar de forma orgánica conceptos de programación y análisis de datos con la genética clínica. Esto ofrece la oportunidad de colaborar entre distintas disciplinas aportándole valor al proyecto.

4.2 OBJETIVOS

La finalidad de este proyecto consiste es desarrollar un modelo de aprendizaje automático que sea capaz de clasificar variantes genéticas relacionadas con cardiopatías hereditarias. La meta es construir una herramienta que logre discriminar eficazmente entre variantes patogénicas y benignas. Esta debe ser capaz de ofrecer una respuesta directa al problema de las variantes inciertas. Para alcanzar este gran objetivo, se han ido estableciendo pequeños objetivos que se han tenido que ir cumpliendo para lograr un trabajo completo.

Los objetivos específicos son:

- Recopilación y preparación del dataset.

Primero se ha buscado determinar el mejor dataset para entrenar nuestro modelo. Se unirán variantes genéticas relacionadas con cardiopatías hereditarias a partir de bases de datos internacionales, usando ClinVar como fuente principal. Para entrenar correctamente el modelo, se debe ejecutar una limpieza dentro de los datos para optimizar el proceso de modelado ya que un dataset mal estructurado y con exceso o falta de datos, tiene un impacto directo sobre el rendimiento del clasificador.

Te sale algo raro en el encabezado de las paginas, como si el titulo del proyecto o del apartado fuera el Anexo I. Revisalo a ver por que te sale esto

- Obtención del mejor modelo.

Se quiere implementar un modelo que sea capaz de categorizar las variantes para determinar en base a sus características si se tratan de variantes patogénicas o benignas. Se evaluará el rendimiento de los modelos mediante técnicas y métricas estándar como la precisión.

- Validación del modelo con fuentes externas.

Para evaluar la capacidad de generalización del modelo se pasará el modelo por datasets externos y se compararan las métricas. Es muy importante cumplir con este objetivo ya que, en el entorno de la salud, las decisiones son claves, y la falta de fiabilidad en las herramientas que utilizan los médicos puede resultar en consecuencias graves para los pacientes.

- Análisis de la interpretabilidad y la utilidad clínica del modelo.

Una vez hemos comprobado que los modelos son robustos y fiables, debemos interpretar los resultados del clasificador. Se analizarán que características tienen más peso o que limitaciones tiene nuestro modelo. También se debe realizar una reflexión analítica sobre la aplicabilidad practica de los modelos en el entorno clínico. En definitiva, analizar si nuestro objetivo final se ha alcanzado.

4.3 METODOLOGÍA

El desarrollo metodológico de este proyecto se ha estructurado en diferentes etapas desde el planteamiento de la cuestión hasta la validación del modelo final. Estas etapas se han definido siguiendo el ciclo habitual de desarrollo de un proyecto de machine learning adaptado a nuestro contexto.

El primer paso consiste en determinar el mejor dataset que usaremos para entrenar al modelo. Este debe ser extenso y tener las variantes suficientes para poder completar nuestro objetivo. Concretamente, se han accedido a bases de datos públicas como ClinVar donde se han obtenido los registros de las variantes genéticas asociadas a cardiopatías. A partir de ahí, se ha realizado un análisis exploratorio inicial del dataset para entender como se han distribuido las muestras y el significado de las variables.

Posterior a eso, se ha realizado la limpieza del dataset. Para ello, atendido a las definiciones de las variables, se han eliminados algunas ya que estas podían quedar definidas por la suma

de otras. También, se han utilizado algoritmos que nos ayudan a determinar la correlación entre variables, ayudándonos a realizar un mejor filtrado en este paso.

Una vez se ha completado la fase de filtrado de datos, empezamos a desarrollar los distintos modelos para encontrar el óptimo para nuestro problema. Para evaluar estos algoritmos nos fijamos en las métricas que nos ofrecen, y para encontrar el mejor modelo, estudiamos su rendimiento mediante visuales orientadas a entender el funcionamiento del modelo.

Una vez realizados todos los modelos y habiendo extraído las métricas de rendimiento, procedemos a realizar una comparación analítica entre los modelos para determinar cual de ellos nos ofrece mejores resultados.

A continuación, llegamos a la última parte técnica del trabajo. Es importante que estos modelos sean generalizables y no se ajusten solo a nuestro dataset. Para ello se usa un conjunto de datos externo. Este pase es de vital importancia ya que nos permite saber si el modelo trabaja adecuadamente con datos que no se encuentran en su conjunto de entrenamiento, asemejándose a una situación real en un hospital.

Finalmente, como en cualquier trabajo de investigación, se realiza un análisis de los resultados y se sacan conclusiones. Para ello seleccionaremos el modelo que mejor resultados nos ha dado, evaluamos su funcionamiento y sacamos conclusiones.

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

4.4.1 PLANIFICACIÓN TEMPORAL

Organizar un trabajo de estas características es clave. Por tanto, al empezar a trabajar en este proyecto, al comienzo del curso académico, se estableció un plan de desarrollo dividiendo el trabajo en fases secuenciales, ya que para empezar una he tenido que haber acabado la anterior, en base a la gestión del tiempo y recursos. Se estableció un plan inicial junto con unos tiempos específicos, pero por la naturaleza de nuestro trabajo podríamos encontrarnos con imprevistos que conllevasen retrasos.

Una vez se decidió el tema del trabajo, las primeras semanas se reservaron para familiarizarse con los conceptos, conocer más del sector, ya sean otros trabajos relacionados con cardiopatías hereditarias o de aprendizaje automático adaptado al sector y plantear nuestro plan de desarrollo.

A partir de este punto, según el cronograma tocaba comenzar con la evaluación de los modelos de clasificación. Se compararon los rendimientos de estos en base a determinadas métricas hasta que se llegó al modelo más robusto. Para realizar los cambios en los modelos se necesitaba conocer su funcionamiento, a que variables les daba más importancia. Para eso se han hecho uso de técnicas de interpretabilidad a medida que se ha ido avanzando en el proyecto.

Conforme se fue avanzando en el proyecto, surgieron nuevas ideas que repercutieron en el plan inicial que tuvo que adaptarse para cubrir estas condiciones. Pese a que el objetivo principal del proyecto no se viese alterado, si se vieron alterados los plazos de las etapas.

Parte de estas ideas pusieron en duda ciertas variables genéticas y poblacionales seleccionadas que a priori no debían aportar poder discriminatorio entre variantes patogénicas y benignas. Esta falta de resolución provocaba un claro riesgo de que el modelo no fuera capaz de generalizar correctamente frente a datos nuevos. Este obstáculo obligó a detener el avance previsto para retornar a la fase de limpieza y revisar exhaustivamente qué características moleculares tenían verdadera relevancia predictiva.

Como resultado de esta dinámica, varias fases del proyecto que inicialmente se habían programado en espacios temporales específicos acabaron solapándose durante el desarrollo del trabajo. Tanto el análisis de matrices de correlación como los análisis de importancia de variables obligaban a iterar en muchas ocasiones los modelos para analizar el desempeño de estos y encontrar posibles mejoras. Este proceso ha sido el que más tiempo ha requerido, ya que era de vital importancia encontrar el modelo que mejor se ajusta a nuestro objetivo.

Como se ha comentado en los dos últimos párrafos hemos usado principalmente dos maneras de discriminar modelos. Uno mediante el estudio de las variables basándonos en nuestro

criterio en base a su significado, y otro usando procesos matemáticos para determinar que variables no son necesarias. Ser capaces de combinar estos dos métodos, probar combinaciones, repetir entrenamientos e ir comprobando los resultados, retrasaron ligeramente el desarrollo del trabajo.

En conclusión, los planes iniciales y finales del proyecto no han sido congruentes. El desarrollo, en lugar de seguir un camino cerrado, se ha fundamentado en un proceso típico de los proyectos de machine learning que incluye ensayo y error, análisis y reajuste metodológico.

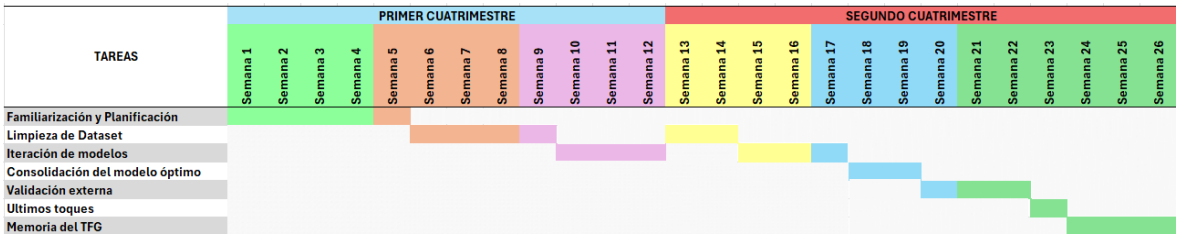

Figura 8 - Planificación temporal al final del trabajo

4.4.2 ESTIMACIÓN ECONÓMICA

Para evaluar la viabilidad de un proyecto al completo, es imprescindible realizar un análisis de costes asociados a su desarrollo. Debido a la naturaleza del proyecto, al tratarse de un proyecto basado íntegramente en software, machine learning, la estructura de costes es más sencilla respecto a cualquier otro proyecto de ingeniería. En este caso, el mayor peso económico recae en las horas de trabajo humano frente a los materiales físicos.

A continuación, se presenta el presupuesto desglosado en tres categorías: costes de personal, costes de hardware y costes de software y licencias.

Como se ha comentado anteriormente, la mayor inversión en este proyecto va ligada al tiempo de desarrollo, por ende, al coste de personal. Para estimar este coste, pese a que se trate de un trabajo de fin de grado, vamos a asumir que el coste laboral del rol de un Ingeniero Junior especializado en análisis de datos asciende a 15€ la hora. Teniendo en cuenta las horas invertidas en la investigación previa, análisis y limpieza de datasets, entrenamiento de modelos, validación externa, conclusiones y la redacción de esta memoria, podemos estimar una dedicación de 450h a este trabajo.

El proyecto se ha desarrollado íntegramente utilizando el dispositivo informático personal. Por ende, al trabajo no se le imputará el valor de compra total del equipo ya que no fue adquirido específicamente para el desarrollo de este proyecto. Sin embargo, podemos hacer una estimación de la amortización correspondiente al tiempo de uso. El dispositivo es un laptop de HP con un procesador i7, 8Gb de RAM y 240 Gb de almacenamiento. Con esta información y las horas de uso en este proyecto, podemos suponer una amortización de 100€ por el uso del dispositivo.

Por último, debemos evaluar los costes de software y licencias. Para el desarrollo de este proyecto se utilizaron herramientas de software gratuito mediante el dispositivo personal. Esta es una de las fortalezas de este trabajo, el uso de herramientas open source lo que no incrementa el coste del proyecto. Todo el desarrollo se realizó en Python utilizando diferentes librerías públicas para la optimización de modelos. A su vez, las bases de datos

de las que se nutre el modelo también son repositorios públicos financiados por instituciones internacionales. El único coste imputable en este apartado al trabajo sería el uso de recursos computacionales en la nube como puede ser Windows, el propio software del dispositivo y desde donde se redacta esta memoria. Sin embargo, el valor amortizado sería mínimo y no es imprescindible ya que se pueden hacer usando otros sistemas operativos.

La siguiente tabla resume el coste total de desarrollo segmentado en las partes explicadas anteriormente.

Nombre	Descripción	Coste estimado
Dispositivo personal	Amortización del equipo personal	100€
Datasets	Bases de datos públicas ClinVar	0€
Software	Windows, Python, …	0€
Costes de personal	Trabajo humano	6.750€
Total	-	6.850€

Tabla 1 - Costes estimados del proyecto.

El coste del proyecto es algo elevado debido a la cantidad de horas dedicada. Sin embargo, la relación coste-beneficio es muy alta al estar hablando de la salud de personas.

Capítulo 5. SISTEMA/MODELO DESARROLLADO

5.1 INTRODUCCIÓN Y SELECCIÓN DEL DATASET

El desarrollo de este proyecto comenzó con una premisa inicial: la identificación y obtención de un conjunto de datos genómicos que reflejen la realidad clínica de las cardiopatías hereditarias. La cantidad de datos almacenada actualmente es inmensa. Sin embargo, no todos están ordenados o resultan adecuados para entrenar modelos de aprendizaje automático, especialmente cuando el objetivo es discriminar la patogenicidad de una variante presente en el diagnóstico médico de una persona. Ahí es donde comienza nuestro proyecto en sí.

Durante una etapa del proyecto, se exploraron diversas bases de datos públicas para encontrar la más adecuada para entrenar nuestro modelo. Aunque repositorios como ClinVar y ClinGen albergan más de 3 millones de registros, los datos en bruto suelen introducir ruido debido a registros incompletos, falta de consenso clínico o mutaciones ajenas a nuestro objetivo. Por ello, seleccionamos exclusivamente ClinVar y aplicamos un filtrado estricto: aislamos únicamente las variantes genéticas asociadas a cardiopatías que contaran con una clasificación de patogenicidad conocida y consensuada. Tras limpiar este ruido y asegurar que cada registro estuviera previamente anotado con las características necesarias para el entrenamiento, obtuvimos un conjunto de datos robusto de 59.000 registros, el escenario ideal para alimentar nuestro clasificador

En estos trabajos se debe determinar una variables objetivo, dependiente o target que debe estar definida en todos los datasets. Así, nuestro clasificador tiene como variable objetivo la patogenicidad de las variantes, pudiendo ser patogénica o benigna. Para conseguir un modelo lo suficientemente robusto, el conjunto de datos debe ser lo suficientemente amplio como para permitir la exclusión de las variantes de significado incierto durante la fase de entrenamiento.

Finalmente, el repositorio seleccionado como pilar del trabajo, que se utilizará para nutrir el modelo, fue Clinvar, un archivo público centralizado mantenido por el *National Center for Biotechnology Information* (NCBI). Esta base de datos se fundamenta en su estatus como el estándar internacional de faacto para el reporte de variantes bajo las directrices del ACMG (*American College of Medical Genetics and Genomics)*, aportándole un valor fundamental para ser el seleccionado para este trabajo. Su arquitectura de datos nos ha permitido aplicar distintas variantes de filtros para quedarnos con un conjunto de datos robusto y suficientemente organizado para comenzar con el desarrollo del modelo que se explicaran en los siguientes apartados.

5.2 DATASET

Como en cualquier trabajo de aprendizaje automático, el pilar clave sobre el que se construye y entrena cualquier modelo predictivo es su base de datos principal. Por ende, el primer paso una vez seleccionado el dataset es revisar sus características. Algunas características ya se habían revisado durante la elección de los datos, pero resulta interesante explorar de una manera más profunda la base de datos que va a nutrir a nuestro modelo para entender mejor el planteamiento del problema.

La base de datos con la que se ha entrenado el modelo consta de 59.377 registros. La estructura de datos que presenta nuestro dataset está compuesta por 31 variables numéricas continuas (tipo *float64*), 1 variable numérica entera (tipo *int64*) y 14 variables categóricas (tipo *object*).

Una de las características que más peso tiene para determinar la calidad de un dataset es la forma en la que están distribuidas las clases. A continuación, la siguiente figura muestra la distribución de clases en nuestro dataset.

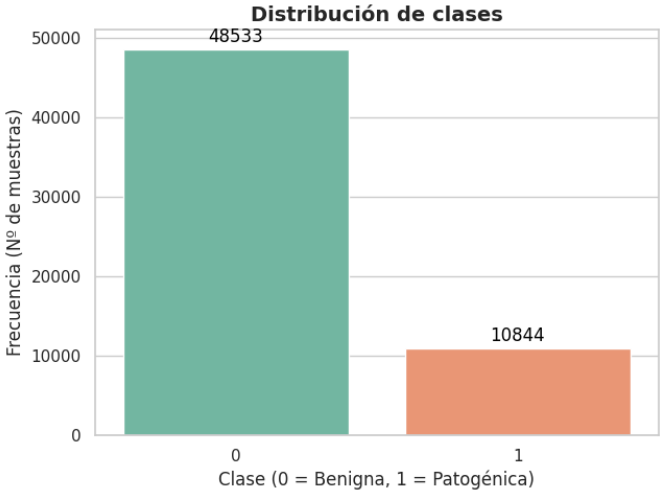

Figura 9 - Distribución de clases en el dataset

Lo más destacable es la diferencia de ejemplos por clases, teniendo una relación de prácticamente 4 ejemplos de clase benigna cada ejemplo por cada ejemplo de clase patogénica. Pese a que pueda resultar un inconveniente de primeras, era previsible que esto fuese a ocurrir. Este fenómeno se debe a que es estadísticamente normal que existan muchas más variantes inofensivas que mutaciones letales.

A continuación, se van a explicar las distintas variables que forman parte de nuestro conjunto de datos agrupadas por tipos.

Identificación y Localización genómica

El primer bloque de variables permite ubicar la variante de forma exacta dentro del genoma de referencia humano (GRCh38) y definir su alteración básica a nivel de secuencia. Para ello, se cuenta con las coordenadas cromosómicas específicas, el nucleótido de referencia y el nucleótido mutado. Adicionalmente, este grupo identifica el gen afectado junto con la nomenclatura estándar internacional, detallando la alteración tanto a nivel de la secuencia de ADN como a nivel de la proteína sintetizada

Consecuencias funcionales y Rutas metabólicas

Un segundo conjunto de características aporta el contexto biológico funcional, describiendo el impacto estructural de la mutación y el rol fisiológico del gen. Variables específicas recogen la predicción del efecto directo de la variante, como puede ser una alteración de sentido erróneo o un cambio en el marco de lectura, e indican si la mutación afecta a un sitio de modificación post-traduccional. Asimismo, se detalla la función general de la proteína codificada y se sitúa la variante dentro de rutas metabólicas concretas utilizando identificadores de bases de datos de referencia biomédica.

Frecuencias poblacionales

La frecuencia con la que una mutación aparece en la población general constituye un criterio clínico determinante, motivo por el cual el conjunto de datos incluye métricas exhaustivas del consorcio gnomAD. Este bloque registra la frecuencia global observada tanto en genomas completos como en exomas, destacando también la frecuencia máxima documentada en cualquiera de las subpoblaciones estudiadas. Además, se contabiliza el número de individuos registrados que presentan la variante en estado de homocigosis o heterocigosis.

Predictores de impacto funcional para los distintos tipos de variantes Otro bloque fundamental está compuesto por predictores computacionales, que consisten en puntuaciones numéricas otorgadas por algoritmos diseñados previamente para evaluar la probabilidad de que la variante resulte dañina. Este grupo abarca desde métricas clásicas y consolidadas en bioinformática hasta puntuaciones de basadas en modelos de aprendizaje profundo. De manera complementaria, se integran evaluaciones probabilísticas que analizan el riesgo de que la mutación altere los sitios de corte y empalme (*splicing*) durante el proceso de maduración del ARN mensajero.

Métricas de Conservación y Tolerancia Genómica

En estrecha relación con lo anterior, se han agrupado diversas métricas de conservación evolutiva y tolerancia genómica. Estas características numéricas evalúan la permisividad funcional de un gen ante la aparición de nuevas mutaciones, incluyendo indicadores de

intolerancia a la variación residual, probabilidades de pérdida de función y puntuaciones que miden cómo de conservada se ha mantenido una secuencia a lo largo de la evolución de las especies.

Evidencia clínica y bases de datos externas

El último grupo de características predictivas vincula la variante genética con la evidencia clínica empírica documentada en estudios médicos previos. Esta sección engloba información sobre asociaciones conocidas con otras enfermedades y códigos ontológicos. También documenta, mediante variables binarias, si la variante ha sido registrada en catálogos de mutaciones somáticas del cáncer, en bases de datos de puntos calientes de mutación o en estudios masivos de asociación a nivel del genoma completo.

Variable Target

Finalmente, dentro de esta extensa dimensionalidad destaca la variable que contiene la clasificación clínica real proporcionada por la plataforma ClinVar. En el marco de nuestro modelo de aprendizaje supervisado, esta columna asume el rol de variable objetivo o dependiente que los algoritmos de clasificación deberán aprender a predecir a partir de la síntesis matemática de todas las características descritas anteriormente.

5.3 MODELO BASE

Una vez tenemos el dataset analizado, podemos comenzar con la fase de desarrollo de modelos. El primer paso dentro de esta fase va a ser la elaboración de un modelo base o baseline. El objetivo de este modelo no es conseguir directamente el algoritmo definitivo para lograr nuestro objetivo, sino establecer un punto de referencia o un umbral mínimo de rendimiento con el cual comparar nuestros siguientes modelos. Es importante tener un baseline en estos proyectos ya que, si las técnicas posteriores, siendo estas más complejas, no logran superar el rendimiento de este modelo, dichas técnicas carecerían de justificación técnica.

Para la elaboración de este primer modelo se quiso mantener el conjunto de datos completamente intacto. El único preprocesamiento aplicado en esta fase fue la imputación de valores nulos para evitar fallos de compilación y a sabiendas que nuestro modelo trabaja adecuadamente con este tipo de valores.

Esta última característica mencionada de nuestro algoritmo supuso un motivo de peso para decantarnos por este, XGBoost. A diferencia de las redes neuronales, XGBoost se ha consolidado como el estándar de la industria para el procesado de datos heterogéneos, destacando su efectividad al trabajar con distintos tipos de datos como variables continuas, categóricas y binarias, como tenemos en nuestra base de datos.

Como se ha comentado anteriormente, se ha seleccionado este algoritmo por su ventaja computacional al poder gestionar adecuadamente la dispersión de datos y los valores faltantes puesto que, en nuestro sector, es muy común que ciertas bases de datos no dispongan de valores para todas las variantes. La forma que tiene XGBoost para trabajar con valores nulos es mediante un algoritmo interno llamado *sparsity-aware split finding* que aprende automáticamente la dirección optima de ramificación minimizando el trabajo del programador al minimizarle la necesidad de aplicar técnicas de imputación.

En segundo lugar, XGBoost presenta una robustez frente al desbalanceo de clases sumamente fiable lo que lo hace más adecuado aún para el proyecto. El modelo construye árboles de decisión de forma secuencial, donde cada nuevo árbol se diseña matemáticamente para minimizar la función de perdida y corregir los errores residuales cometidos por los anteriores. Dentro de este algoritmo existen hiperparámetros que permiten ajustar el peso de las predicciones en favor la clase con menos presencia lo que equilibra la salida del modelo.

Una vez explicado el motivo de la elección de este modelo, comenzamos con su programación y la obtención de los resultados del modelo base.

Accuracy: 0.9869 *Balanced Accuracy: 0.9720*

Una vez obtenidos estos resultados, procedemos a analizarlos y sacar conclusiones para ir optimizando el modelo. Para plantear posibles mejoras en el modelo, hay que entender cómo funciona por dentro, en que se ha basado el modelo para tomar las decisiones. Para ello, hacemos uso de la herramienta SHAP. SHAP nos permite desglosar y cuantificar la contribución exacta de cada variable en la decisión final del algoritmo, en definitiva, un análisis *feature importance*. (Lundberg, 2017) SHAP nos permite saber varias características. Primero nos permite cuantificar el peso que tiene una variable dentro del modelo. Además, podemos saber cómo los valores específicos de una mutación en un paciente concreto empujan la predicción hacia la categoría patogénica o benigna. De esta forma, podemos entender como razona nuestro modelo para ir optimizándolo hasta encontrar el más adecuado.

Sobre el modelo inicial realizamos un análisis SHAP para ver posibles variables prescindibles en nuestro

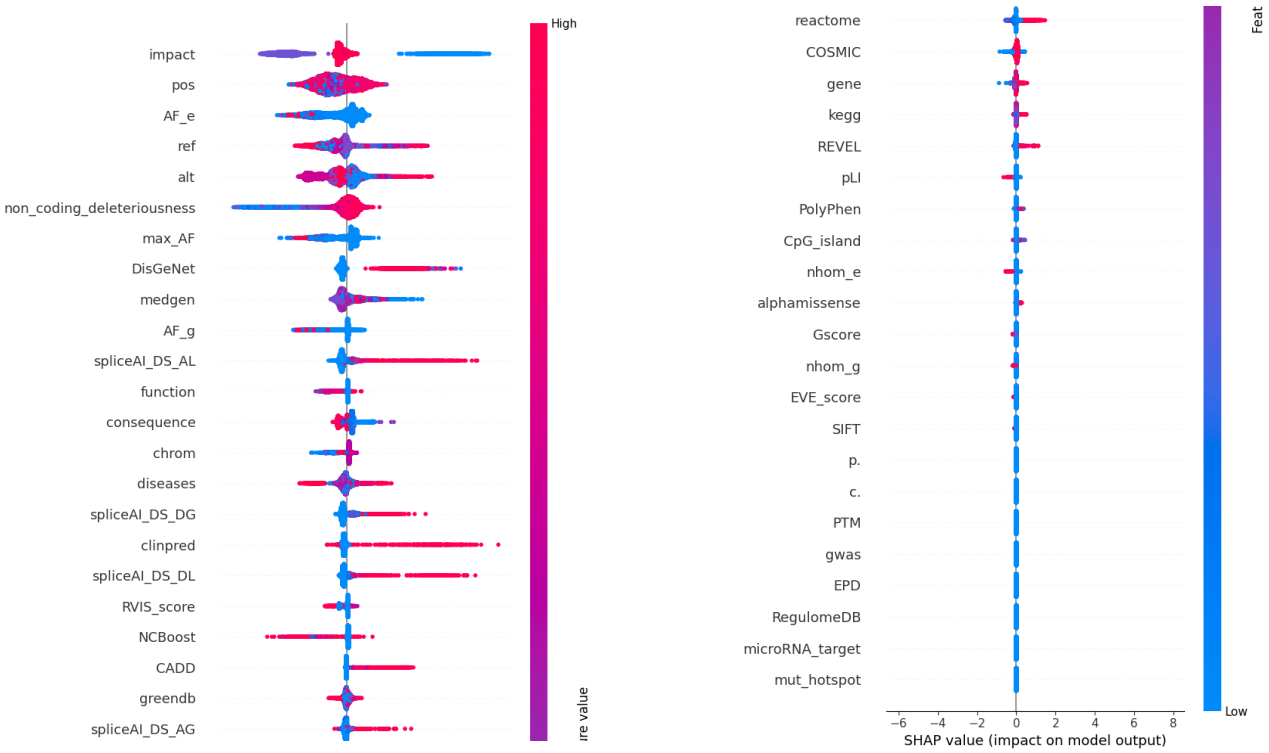

Figura 10 - Feature Importance SHAP modelo inicial.

La manera de interpretar este grafico es la siguiente. Para empezar, las variables estan ordenadas de las que mayor importancia dota el modelo, a la que menos. Los puntos en cada variable representan un ejemplo del dataset. El color del punto es el valor que toma la variable en ese ejemplo. De esa forma, un punto rojo en una variable, hace referencia a un valor muy elevado de la variable en el ejemplo que representa ese punto. Por otro lado, el eje horizontal representa el peso que tiene la variable para determinar la clase del ejemplo. Los puntos que estan situados a la derecha tienen más probabilidad de ser patogenicos, y los que estan situados más a la izquierda tienen más probabilidad de ser benignos. Una vez entendido esto, vamos a fijarnos en las variables y a analizar el resultado.

La variable *Impact* es la variable con mayor importancia. Podemos ver como valores bajos de esta variable pesan mucho para determinar esa variante como patogénica. Por otro lado, los valores altos en esta variable no aportan apenas información sobre la patogenicidad de la variable. Y valores intermedios empujan más a que la decisión final sobre la clase del ejemplo sea benigna. Posteriormente se realizará un análisis exhaustivo de las variables para determinar bajo criterio clínico si SHAP está dando el peso adecuado a las variables adecuadas. Ya que veremos, que no siempre está en lo correcto y que siempre se debe aportar juicio humano en estos proyectos.

En la cola del gráfico, vemos como hay valores que no aportan información. Estas probablemente sean variables sin ejemplos, o con muy poca cantidad de valores no nulos. La clave para optimizar el modelo es averiguar de que variables prescindir y cuales tienen más valor. Una vez realizado el modelo inicial sobre el que poder comparar el resto de los modelos, vamos a comenzar a optimizarlo.

Para ello, se ha trabajado separando dos líneas de desarrollo, dándole más importancia a la segunda. En la primera línea de trabajo se han usado únicamente métodos analíticos y matemáticos para discriminar variables, ya sean matrices de correlaciones o valores de importancia en los modelos. Y en la segunda, se ha realizado un filtrado más exhaustivo atendiendo al criterio clínico para eliminar variables de una manera más subjetiva, combinándolo a su vez con métodos matemáticos.

5.4 PRIMERA LÍNEA DE TRABAJO

Tras comprobar las limitaciones del modelo base, con variables que no aportaban valor alguno, la primera estrategia metodológica adoptada para optimizar el rendimiento del modelo se fundamentó exclusivamente en métodos analíticos y matemáticos. El objetivo de esta línea de trabajo era reducir la dimensionalidad del conjunto de datos de forma objetiva eliminando variables que no aportaran información adicional, que introdujeran redundancia o ruido, sin depender del conocimiento médico. Para ello se han seguido una serie de pasos que quedan detallados a continuación.

Primero, y como en cualquier proyecto de machine learning, se debe realizar un pre-procesado del dataset. Este consistió en suprimir toda la información que actuaba como una "matrícula" o identificador único de la variante. Esto incluyó las coordenadas genómicas exactas *chrom* y *pos*, los nucleótidos implicados *ref* y *alt* y el nombre del gen afectado *gene*. Mantener estas variables introducía un riesgo de sesgo o *overfitting* por memorización. Si el modelo disponía de la posición exacta o del nombre del gen, existía la alta probabilidad de que no aprendiera las reglas bioquímicas que hacen dañina a una mutación, sino que simplemente memorizara qué coordenadas específicas estaban etiquetadas como patogénicas en el conjunto de entrenamiento. Eliminar estos identificadores fuerza al modelo a generalizar basándose estrictamente en las propiedades predictivas, garantizando su robustez ante variantes o genes que nunca haya procesado.

El primer filtro cuantitativo aplicado consistió en eliminar las variables que no tenían más de un 0.2% de ejemplos puesto que se consideran variables que no aportan valor alguno al modelo, lo ralentiza y reduce su eficiencia. Si una variable genética o poblacional solo contiene información empírica en una fracción tan minúscula de los casos, estando vacía o siendo constante en el 99.8% restante, carece por completo de la masa crítica necesaria para que el algoritmo pueda extraer un patrón generalizable. Las variables que cumplen estas características son *RegulomeDB, EPD, PTM, gwas, microRNA_target, mut hotspot, c.* y *p.*.

Una vez realizado el modelo toca evaluar el rendimiento del modelo. Posteriormente compararlo con el modelo inicial y analizar posibles mejoras en este. Los resultados del modelo son los siguientes:

Accuracy: 0.9869 *Balanced Accuracy: 0.9720*

El siguiente paso metodológico consistió en delegar la discriminación de características al algoritmo SHAP. Los resultados de modelo son los siguientes:

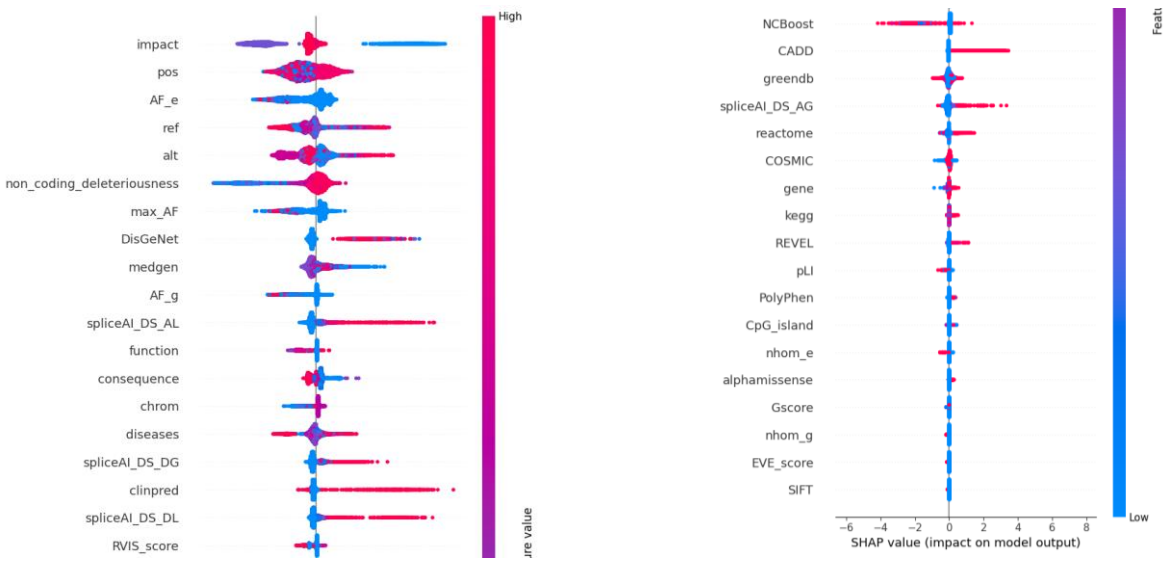

Figura 11 - Feature importance SHAP del primer modelo.

Con los resultados ya representados, pasamos a la fase de análisis. Como se comentó en la explicación del algoritmo SHAP, el orden de las variables está ordenadas de mayor importancia a menor. Para optimizar el modelo, se han de eliminar variables que no aporten suficiente valor al modelo. Para lograr este objetivo, se va a ir cambiando el umbral establecido hasta lograr el modelo con mayor rendimiento.

Para el primer algoritmo desarrollado para mejorar el actual, el umbral se establece por debajo de la variable **pLI** (*score de tolerancia a loss of function*), suponiendo que las

variables con menos importancia no son relevantes para el modelo. Los resultados de este modelo son los siguientes:

Accuracy: 0.9869 *Balanced Accuracy: 0.9720*

Respecto al modelo anterior (*Accuracy: 0.9869, Balanced Accuracy: 0.9720*) no observamos mejora alguna. A la hora de analizar y comparar resultados, es importante mirar la precisión y la precisión balanceada, especialmente en nuestro caso. La precisión balanceada (*Balanced Accuracy*) es una precisión que se basa no únicamente en la precisión general del modelo, sino que tiene en cuenta también la cantidad de ejemplos por clase para ajustar esta métrica. Debido a las características de nuestro dataset, al estar las clases desbalanceadas, resulta mucho más interesante fijarnos en la precisión balanceada antes que en la precisión general. Una vez explicado esto y comparando los resultados obtenidos, podemos ajustar más el umbral buscando la mejora del modelo. El siguiente modelo que probamos es discriminando las variables con menos importancia que **gene**. Y los resultados obtenidos son los siguientes:

Accuracy: 0.9880 *Balanced Accuracy: 0.9747*

A diferencia del anterior modelo, ajustando el umbral y haciéndolo más estricto, se ha mejorado notablemente tanto la precisión, como la precisión balanceada. Sin embargo, no sabemos si estamos ante el modelo más optimizado de esta línea de desarrollo. Para averiguarlo, debemos seguir iterando los modelos, estableciendo diferentes limites dentro de las variables y comparar los resultados.

Una vez sabemos que estableciendo el umbral en *gene* se ha mejorado el modelo, probamos subiendo y bajando el umbral un nivel para corroborar que nos encontramos ante el modelo más optimo. Tanto si subimos el límite a la variable superior de gene como si bajamos el límite un nivel, los resultados se ven empeorados (*Accuracy: 0.9875, Balanced Accuracy: 0.9733* y *Accuracy: 0.9867, Balanced Accuracy: 0.9737* respectivamente).

Por último, como alternativa para eliminar variables, se ha recurrido a la valuación de la multicolinealidad entre las variables numéricas continuas del conjunto de datos. En el contexto del aprendizaje automático, cuando dos o más características están altamente

correlacionadas, aportan esencialmente la misma información al algoritmo predictivo. Esta redundancia no solo incrementa innecesariamente el coste computacional del entrenamiento, sino que puede inestabilizar ciertos modelos y diluir el peso real de otras variables más relevantes.

Para identificar este solapamiento informativo, se calculó una matriz de correlación. Este análisis matemático bidimensional permitió detectar de forma precisa pares de predictores o métricas poblacionales que exhibían una fuerte dependencia lineal. Ante la detección de estas agrupaciones redundantes, se estableció un criterio objetivo de exclusión: mantener únicamente la variable que mostrara una mayor correlación individual con la variable objetivo (clinvar), descartando matemáticamente el resto de las características colineales para simplificar la arquitectura del modelo sin sacrificar su poder predictivo.

La matriz de correlaciones de las variables que se han mantenido en el modelo es la siguiente:

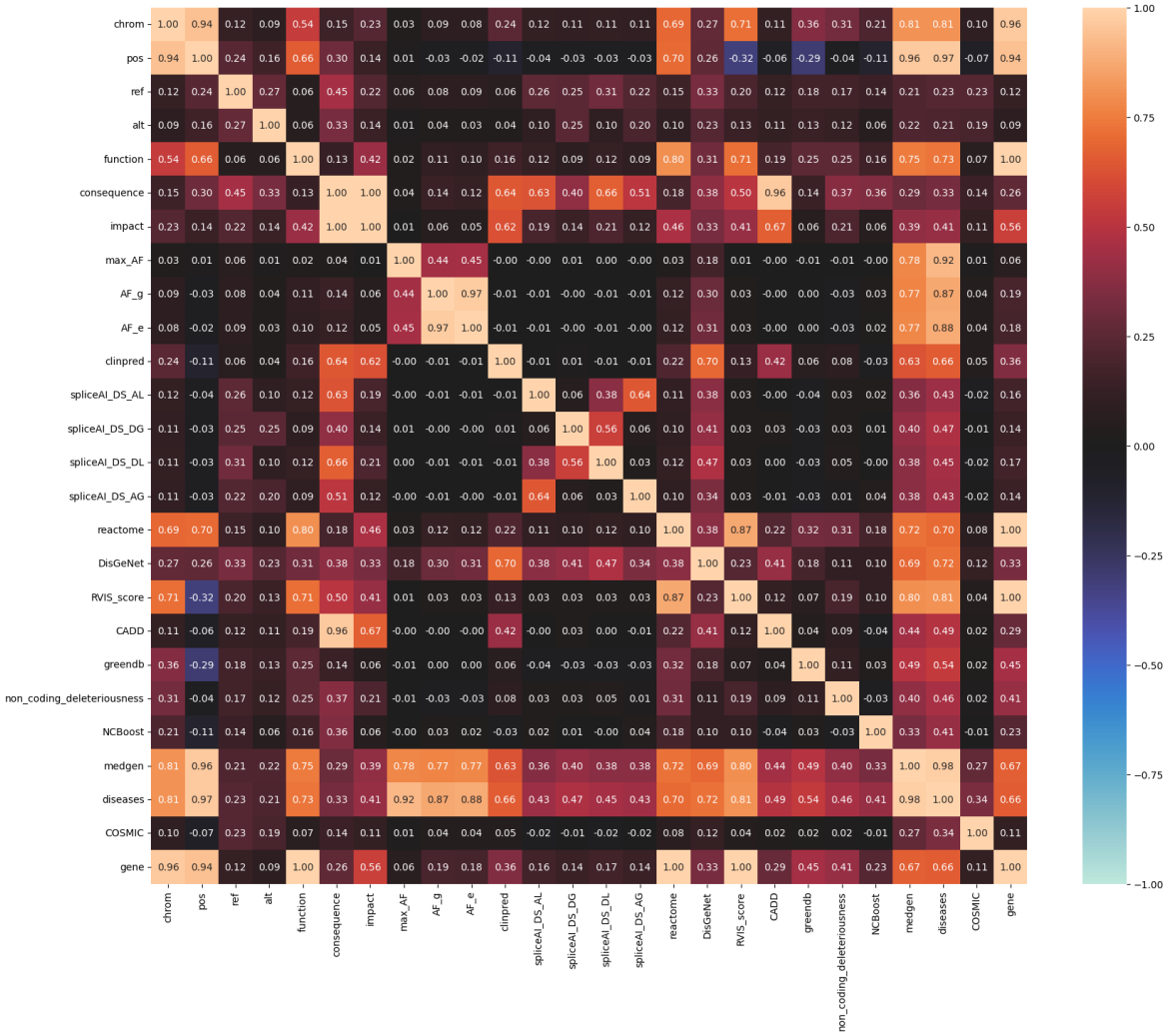

Figura 12 - Matriz de correlaciones de la primera línea de trabajo.

La matriz de correlación generada ha permitido identificar y cuantificar las relaciones lineales subyacentes entre las variables del conjunto de datos. En el mapa de calor, los valores cercanos a 1.00 indican una correlación positiva perfecta, lo que en el contexto del aprendizaje automático se traduce en redundancia informativa o multicolinealidad.

Se observa una correlación perfecta de **1.00** entre las variables consequence e impact. Esto indica una colinealidad estricta derivada de la propia naturaleza bioinformática de los datos: la consecuencia de la mutación dictamina directamente su nivel de impacto. Mantener ambas variables en el modelo es matemáticamente redundante, por lo que se debería poder prescindir de una para optimizar el modelo.

Una vez analizadas las correlaciones entre las variables, se han creado distintos modelos buscando alternativas para sustituir variables con altas correlaciones. El mejor resultado obtenido se ha encontrado prescindiendo de la variable *pos* y los resultados han sido los siguientes:

Accuracy: 0.9842 *Balanced Accuracy: 0.9671*

Estos resultados no mejoran el modelo obtenido anteriormente. Por tanto, este cambio no aporta valor al algoritmo por lo que no es necesario realizarlo, y nos mantenemos con el modelo anterior.

Una vez elaborados y evaluados todos estos modelos, podemos afirmar con seguridad que hemos logrado el modelo óptimo bajo el requerimiento de usar únicamente métodos matemáticos. Ahora bien, ¿es este modelo realista, aplicable a la vida real, consecuente? Para responder a esta pregunta, se ha elaborado la segunda línea de desarrollo la cual va a trabajar bajo el criterio clínico y tomando decisiones en base al juicio humano.

5.5 SEGUNDA LÍNEA DE TRABAJO

Frente al enfoque puramente cuantitativo y estadístico desarrollado en la primera línea de trabajo, la segunda estrategia metodológica se fundamentó en el conocimiento del dominio biológico, la semántica de los datos y el juicio crítico investigador. El objetivo de esta fase fue purgar el conjunto de datos identificando y eliminando aquellas variables que, independientemente de sus métricas de correlación matemática, carecían de sentido lógico o utilidad clínica para resolver el problema de clasificación planteado.

Como en la primera línea, el preprocesado del dataset se debe realizar lo primero en este tipo de proyectos para poder trabajar con datos controlados y que los modelos sean concluyentes.

Como análisis más profundo, se retiraron las variables de texto libre asociadas a la descripción biológica general, tales como el papel que tenía la propia proteína *function* y los identificadores de rutas metabólicas *kegg* y *reactome*. Dado que estas variables describen el comportamiento del gen sano y no el impacto particular de la mutación, carecen de poder

discriminatorio. Por ejemplo, dos variantes en el mismo gen, una benigna y otra patogénica, compartirían exactamente el mismo texto. En esta misma línea de limpieza de ruido, se descartó la variable CpG_island, al considerarse que, para el contexto específico de las patologías cardiovasculares estudiadas, su inclusión aportaba más dispersión que valor predictivo real.

Otro criterio cualitativo aplicado para mejorar el modelo abordó la eficiencia de los predictores y las frecuencias poblacionales. Basándose en el conocimiento del estado del arte en genómica, se decidió prescindir de herramientas predictivas clásicas como SIFT y PolyPhen. La justificación clínica radica en que el dataset ya incorpora predictores de última generación que internamente ya agrupan o superan ampliamente el rendimiento de estos algoritmos más antiguos, haciendo que su presencia sea redundante y computacionalmente ineficiente. De manera análoga, se eliminó la variable categórica *consequence*, al entenderse que su información biológica ya se encontraba sintetizada y graduada en otras métricas de impacto más manejables para el algoritmo. Por último, se retiró la frecuencia poblacional máxima (max_AF), optando por mantener exclusivamente las frecuencias globales (AF_g, AF_e) para evitar solapamientos informativos y simplificar la toma de decisiones del modelo.

Una vez se han eliminado las variables que no aportan información alguna, toca evaluar el modelo para comprobar su eficacia. Los resultados del modelo inicial sin optimizar únicamente son los siguientes:

Accuracy: 0.9660 *Balanced Accuracy: 0.9280*

Los resultados son muy positivos, frente a ser haber dismimuido la precisión con el modelo calculado en la primera linea. ¿A que se puede deber este cambio? Es un fenómeno extremadamente común en proyectos de Machine Learning especialmente aplicado a la salud y, de hecho, desde una perspectiva de ingeniería clínica, es una excelente señal metodológica. Que el modelo obtenga métricas ligeramente inferiores al modelo puramente matemático no significa que sea un modelo peor. Al contrario, es muy probable que sea un

modelo mucho más seguro y realista. Esta ligera caída en el rendimiento se puede justificar por varios motivos técnicos fundamentales.

El modelo matemático original se beneficiaba enormemente del sobreajuste por memorización. Al disponer de identificadores únicos como el nombre del gen, las coordenadas genómicas exactas o los códigos de enfermedades previas, el algoritmo disponía de "atajos" estadísticos. XGBoost es un clasificador matemático implacable. Si detecta que la aparición de un gen concreto coincide repetidamente con la etiqueta patogénica en tu conjunto de entrenamiento, simplemente memoriza esa relación estadística en lugar de aprender las verdaderas reglas bioquímicas. Al aplicar un filtro clínico y eliminar estos atajos, obtenemos modelos que enfrentan el problema de una manera más realista. Es completamente lógico que el rendimiento estadístico caiga ligeramente, porque la tarea a la que se enfrenta ahora el algoritmo es genuinamente más difícil, pero mucho más fiel a un escenario de diagnóstico real.

Los enfoques basados unicamente en métodos matemáticos, tienden a explotar correlaciones casuales o señales de ruido muy débiles que casualmente cuadran en el conjunto de entrenamiento, pero que biológicamente carecen de sentido causal. Al suprimir este ruido mediante nuestro criterio, se han sacrificado conscientemente unas décimas de precisión bruta a cambio de ganar una interpretabilidad y una generalización absolutas. En el ámbito de la medicina genómica, un modelo que obtiene un rendimiento ligeramente menor, pero que toma sus decisiones por los motivos biológicos correctos, es infinitamente superior a un modelo que acierta un poco más, pero lo hace apoyándose en sesgos y atajos matemáticos.

Una vez desarrollado este modelo, se procedió a evaluar el peso predictivo real de las variables restantes. Para comprender la manera de evaluar el modelo y analizar posibles cambios del modelo para su optimización, se ejecuta el algoritmo SHAP que se encarga de evaluar la importancia de las variables en el modelo. La visualización de resumen resultante revela de forma explícita la jerarquía de importancia de las variables dentro del modelo XGBoost, permitiendo observar no solo qué características eran las más determinantes, sino cómo el valor de cada una empuja la predicción hacia la clase patogénica o benigna.

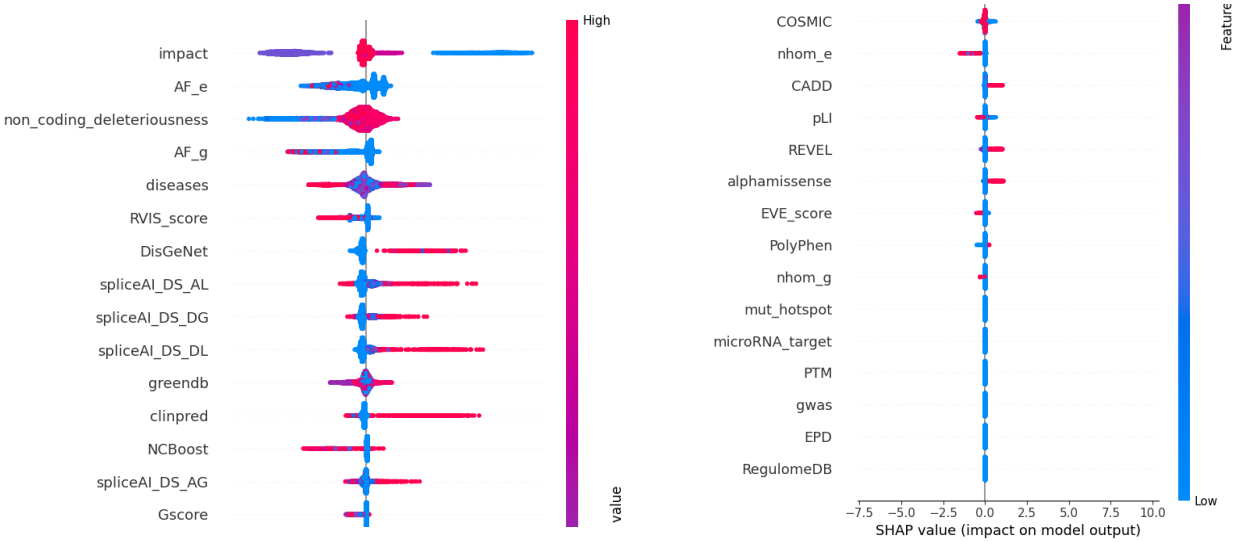

Figura 13 - Análisis SHAP del primer modelo de la segunda línea de trabajo.

El gráfico SHAP de este modelo muestra una clara polarización en la utilidad de los datos. En la parte superior, características como el impacto funcional (*impact*), *non_coding_deleteriousness* y las frecuencias alélicas poblacionales (*AF_e* y *AF_g*) demostraron tener una dispersión horizontal masiva. Esto confirma que estas variables son los verdaderos indicadores de inferencia del sistema, teniendo un gran impacto en la salida del modelo. Por el contrario, el tercio inferior del gráfico mostró una extensa cola de variables, tales como la *microRNA_target*, puntos calientes mutacionales (*mut_hotspot*), regiones promotoras (*EPD*) o bases de datos regulatorias (*RegulomeDB*), cuyos valores de se concentraban estrictamente en torno al cero, demostrando un impacto muy leve o incluso nulo sobre las fronteras de decisión del algoritmo.

Fundamentándose en esta evidencia matemática y visual, se diseñó un proceso de optimización iterativa basado en la eliminación de variables menos importantes. Las variables situadas en la parte inferior del espectro SHAP fueron suprimidas progresivamente del conjunto de entrenamiento al considerarse ruido estadístico ineficiente. Tras cada ciclo de eliminación, el modelo predictivo fue reentrenado, ejecutando en paralelo nuevas búsquedas para reajustar los hiperparámetros a la nueva dimensionalidad del *dataset*. Esta

metodología de ensayo, purga y reentrenamiento permitió probar de forma empírica múltiples combinaciones de variables.

Estos modelos se fueron mejorando hasta alcanzar el modelo con mejores métricas. Este modelo se obtuvo tras eliminar las variables *pLI, REVEL*, *alphamissense*, EVE_*score*, *PolyPhen*, *nhom_g*, *mut_hotspot*, *microRNA_target*, *PTM*, *gwas*, *EPD* y *RegulomeDB*.

Tras reajustar los hiperparámetros sobre este conjunto de datos purificado, la evaluación del modelo arrojó unos resultados mejores respecto al primer modelo. Las métricas obtenidas en este modelo fueron:

Accuracy: 0.9671 *Balanced Accuracy: 0.9301*

El sistema alcanzó una precision del 0.9671, mejorando ligeramente el modelo anterior. Sin embargo, dada la presencia de un desbalanceo de clases en el *dataset* original, la exactitud global puede resultar una métrica optimista como se comentó anteriormente. Por este motivo, el verdadero indicador del éxito de esta optimización cuantitativa reside en la precisión balanceada la cual aumentó a un 0.9301.

Alcanzar una precisión balanceada de 93% tiene una implicación clínica y técnica fundamental. Esta métrica se calcula como la media aritmética entre la sensibilidad y la especificidad. Por tanto, este resultado demuestra que el modelo no ha caído en la paradoja de la exactitud, prediciendo sistemáticamente la clase mayoritaria, sino que ha aprendido con éxito a discriminar la clase minoritaria.

Tras concluir las sucesivas iteraciones de preprocesamiento, el filtro basado en criterios tanto matemáticos como clínicos, y el ajuste exhaustivo de hiperparámetros, se ha logrado consolidar la arquitectura definitiva del modelo predictivo. El algoritmo XGBoost resultante ha demostrado, en sus métricas globales preliminares, una alta capacidad para lidiar con la dispersión de la información y el severo desbalanceo intrínseco de las clases genéticas. No obstante, en el ámbito de la ingeniería aplicada a la salud, la fiabilidad de un sistema de

soporte al diagnóstico no puede justificarse basándose únicamente en métricas de rendimiento estadístico estáticas.

Por consiguiente, el siguiente capítulo de este trabajo se dedicará al Análisis Detallado de Resultados. En él, se procederá a deconstruir el comportamiento interno del clasificador mediante representaciones gráficas avanzadas. Se evaluará su capacidad de discriminación real frente a la clase minoritaria y la Matriz de Confusión Normalizada, y se demostrará su auditabilidad clínica paciente a paciente empleando técnicas de para desenvolver el funcionamiento del modelo.

Finalmente, dado que el riesgo de sobreajuste es el mayor desafío en los modelos de base de datos única, el rendimiento del algoritmo no se considerará clínicamente validado hasta superar su prueba definitiva. Como culminación de este proyecto investigativo, el modelo optimizado será expuesto a dos conjuntos de datos externos e independientes. La evaluación del sistema frente a cohorte validación externa será el indicador definitivo que confirme si el algoritmo ha logrado generalizar las reglas biológicas de la patogenicidad, demostrando su viabilidad técnica para ser implementado en un entorno médico real.

Capítulo 6. ANÁLISIS DE RESULTADOS

En este capítulo se presenta la evaluación exhaustiva del modelo predictivo desarrollado, trasladando el enfoque desde la optimización algorítmica hacia la validación de su utilidad diagnóstica. Como se explicó durante la fase de diseño, el rendimiento de un sistema de aprendizaje automático aplicado a la genómica cardiovascular no puede medirse exclusivamente mediante métricas de precisión general, especialmente ante la presencia de un desequilibrio de clases tan pronunciado. Por ende, la evaluación empírica de los resultados se estructurará en dos dimensiones fundamentales.

En la primera fase, se diseccionará el comportamiento interno y la capacidad discriminativa del modelo optimizado. Para ello, se emplearán herramientas analíticas avanzadas, tales como la Matriz de Confusión Normalizada, que permitirán cuantificar de forma objetiva la sensibilidad clínica del sistema frente a la clase patogénica minoritaria. Asimismo, se integrará el marco analítico SHAP para entender la forma de razonar del modelo y así, garantizar la auditabilidad médica de los resultados.

En la segunda fase, la arquitectura final será sometida a una prueba necesaria y definitiva dentro de los trabajos de machine learning: la evaluación frente a un conjunto de datos externo. Este set de datos independiente, completamente invisible para el algoritmo durante su ciclo de entrenamiento, determinará de forma concluyente si el modelo ha logrado generalizar con éxito las reglas subyacentes de la biología molecular, confirmando su robustez y viabilidad técnica para ser desplegado en entornos clínicos reales.

6.1 EVALUACIÓN DEL RENDIMIENTO DEL MODELO

En este apartado, se detallará la evaluación del rendimiento calculado del modelo optimizado. Primero es necesario analizar el modelo para entender cómo funciona por dentro. Esto facilita a los profesionales a entender que variables son las más importantes a

la hora de determinar la patogenicidad de las mutaciones. Para ello, vamos a recurrir al algoritmo SHAP que nos mostrará las variables en orden de importancia.

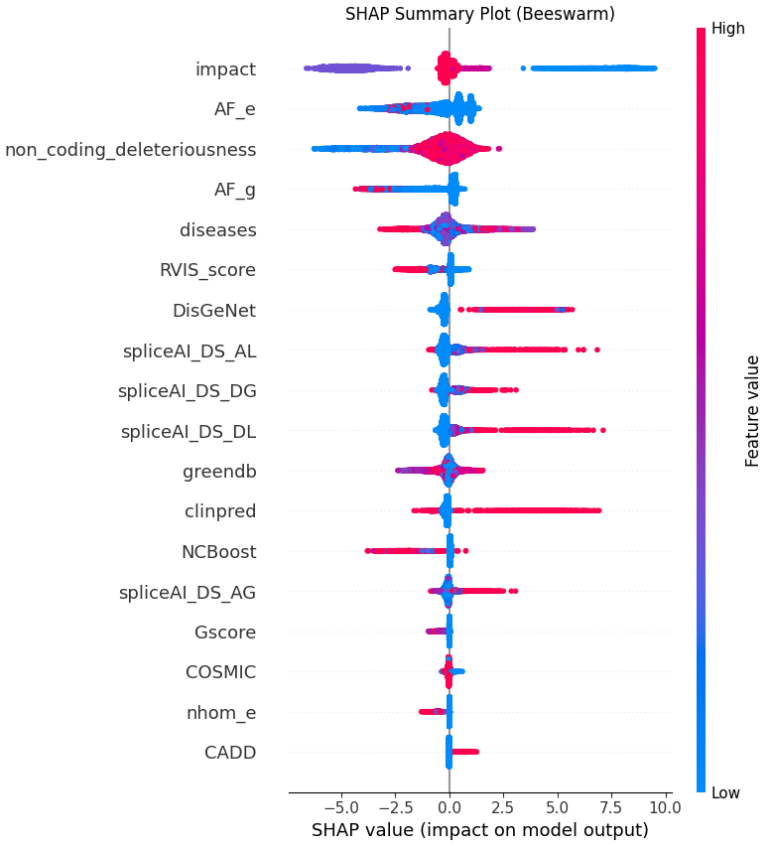

Figura 14 - Análisis SHAP del modelo final.

En la cima de columna de variables, se sitúa la variable impact. Su dispersión horizontal demuestra que es el factor más determinante para separar ambas clases. Inmediatamente después, destaca el peso de *AF_e* y *non_coding_deleteriousness*. En esta métrica, se observa claramente cómo los valores bajos (puntos azules) empujan con fuerza la predicción hacia la izquierda, inclinando al algoritmo a clasificarla como benigna. Esto indica que, si el algoritmo detecta que la mutación no altera regiones no codificantes críticas, descarta casi automáticamente el riesgo patogénico, actuando como un excelente filtro de seguridad.

En el tercio medio del gráfico, encontramos la batería de puntuaciones computacionales, *clinpred* y *NCBoost,* y muy especialmente, los predictores de alteraciones en el corte y

empalme del ARN, *spliceAI_DS_AL, spliceAI_DS_DG* y *spliceAI_DS_DL*. El comportamiento de estas variables es asimétrico y se puede destacar lo siguiente: los valores bajos se concentran en una densa línea vertical sobre el cero, indicando que un *splicing* normal no aporta información definitiva. Sin embargo, cuando estos valores son altos, indicando una alta probabilidad de rotura del sitio de *splicing* o un alto score clínico, los valores SHAP se disparan drásticamente hacia la derecha. Esto significa que el algoritmo utiliza estas herramientas como "señales de alarma". No las necesita para decir que alguien está sano, pero cuando detectan una anomalía estructural, son capaces de dictaminar por sí solas la patogenicidad de la variante.

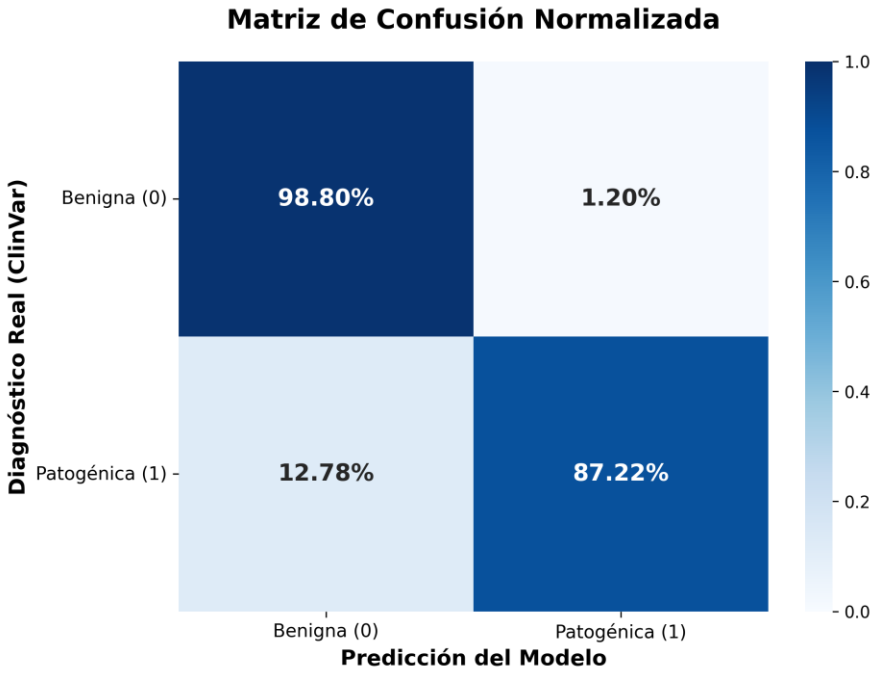

Figura 15 - Matriz de confusión del modelo final.

Para evaluar de forma objetiva la capacidad de generalización del modelo frente al desequilibrio de clases documentado en el *dataset* original, 81.7% benignas frente a 18.3% patogénicas, se ha analizado la Matriz de Confusión Normalizada. Esta representación visual resulta fundamental, ya que, al expresar los resultados en porcentajes relativos a cada clase real, se elimina el sesgo provocado por la gran cantidad de muestras de la categoría

mayoritaria, permitiendo evaluar el rendimiento intrínseco del algoritmo en ambos escenarios clínicos.

En el cuadrante superior izquierdo, se observa que el modelo logra identificar correctamente el 98.80% de las variantes genéticas benignas. Clínicamente, esto se traduce en una Especificidad sobresaliente y en una tasa de Falsos Positivos extraordinariamente baja de 1.20% concretamente. En la práctica del diagnóstico genómico, este alto grado de precisión es vital para evitar el sobrediagnóstico, garantizando que no se alarme innecesariamente a pacientes sanos ni se les someta a seguimientos cardiológicos preventivos o intervenciones terapéuticas que no necesitan.

El verdadero reto del proyecto residía en la detección de la clase minoritaria. Como muestra el cuadrante inferior derecho, el algoritmo es capaz de clasificar correctamente el 87.22% de las mutaciones que son genuinamente patogénicas. Este resultado empírico de Sensibilidad demuestra el éxito de las técnicas de optimización y de los pesos penalizadores implementados en la arquitectura del modelo. El modelo ha evitado la eficazmente aprender los resultados por memorización, y ha aprendido las reglas bioquímicas que definen la patogenicidad de una variante, logrando capturar a la gran mayoría de los pacientes en situación de riesgo.

Desde una perspectiva crítica, es necesario abordar el 12.78% de variantes patogénicas que el modelo clasifica erróneamente como benignas cuadrante inferior izquierdo. En medicina predictiva, este es el error más costoso, puesto que implica no detectar un riesgo real. Los falsos positivos son fácilmente descartables después con una prueba complementaria, que actualmente siempre se realiza, puesto que ningún modelo predictivo se utiliza como evidencia única de diagnóstico por ahora.

Sin embargo, en el contexto de las predicciones basadas en Machine Learning, alcanzar casi un 90% de detección sin la ayuda de ensayos funcionales o historiales clínicos directos es un resultado correcto y asumible. La existencia de esta pequeña tasa de error justifica la concepción de este modelo no como un sustituto del genetista, sino como una potente herramienta o soporte a la decisión (CDSS - *Clinical Decision Support System*), capaz de

filtrar el 98% del "ruido" genómico benigno y priorizar los casos dudosos para su revisión manual por parte de un panel de expertos.

Como complemento al análisis estático proporcionado por la Matriz de Confusión, resulta indispensable evaluar la robustez del modelo independientemente del umbral de probabilidad elegido para tomar la decisión final. Para este propósito, se introduce el análisis mediante la Curva de Característica Operativa del Receptor, Curva ROC, y el cálculo de su respectivo Área Bajo la Curva, AUC.

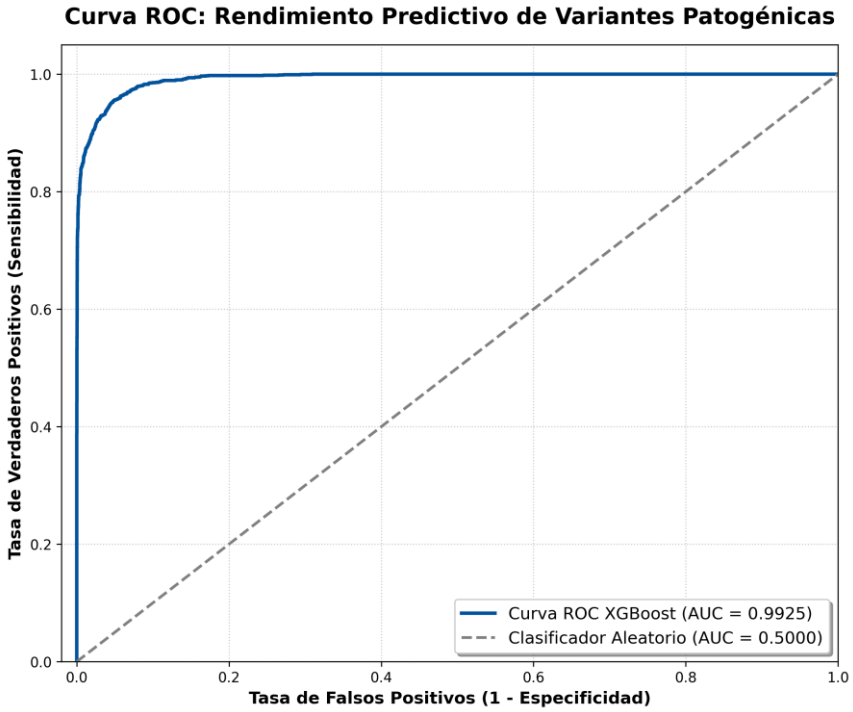

Figura 16 - Curva ROC del modelo final.

La evaluación de la robustez del modelo se ilustra mediante el análisis de la Curva ROC (*Receiver Operating Characteristic*), una herramienta gráfica fundamental para evaluar sistemas de clasificación binaria. La curva ROC es una representación gráfica que evalúa el rendimiento predictivo de un modelo de clasificación binaria a través de todos los umbrales de decisión matemáticos posibles. En lugar de fijar un único punto de corte para decidir si una variante es patogénica o benigna, la curva traza el compromiso dinámico entre la Sensibilidad, la Tasa de Verdaderos Positivos, representada en el eje Y, y la Tasa de Falsos

Positivos, 1 - Especificidad, en el eje X. A medida que el algoritmo se vuelve más permisivo o estricto en sus predicciones, la gráfica ilustra cómo varía su capacidad para detectar mutaciones dañinas frente al coste de emitir falsas alarmas. Un modelo altamente eficaz dibujará una curva que se arquea rápidamente hacia la esquina superior izquierda del plano, lo que indica que es capaz de identificar casi todos los casos positivos reales incurriendo en muy pocos errores.

Para cuantificar objetivamente este comportamiento visual, se utiliza el AUC (*Area Under the Curve*), que mide la superficie bidimensional situada por debajo de la curva ROC, tomando valores entre 0.0 y 1.0. Un AUC de 0.5 representa la línea diagonal central, indicando un modelo con nula capacidad discriminativa que opera al nivel del puro azar. Por el contrario, un AUC cercano a 1.0 —como el obtenido en este proyecto— certifica una separación casi perfecta entre clases. En entornos de genómica médica con un severo desequilibrio de clases, el AUC se establece como la métrica más robusta y fiable, ya que, a diferencia de la precisión general, evalúa la capacidad intrínseca del algoritmo para distinguir la clase minoritaria sin dejarse engañar por la predominancia estadística de la clase mayoritaria.

Como se evidencia en la representación visual, el desempeño del algoritmo destaca por alcanzar un Área Bajo la Curva de 0.9925. En el contexto del aprendizaje automático, donde el clasificador aleatorio base se sitúa en un AUC de 0.5000, obtener un valor tan próximo a la unidad ideal demuestra una capacidad discriminativa excepcional para separar matemáticamente las variantes patogénicas de las benignas.

Desde una perspectiva geométrica, la superioridad predictiva del modelo se refleja en la pronunciada concavidad de la curva principal. El trazo asciende de manera casi vertical hacia la esquina superior izquierda del plano desde los primeros umbrales de decisión. Esta trayectoria nos indica que el algoritmo es capaz de alcanzar una tasa muy buena de verdaderos positivos (sensibilidad), superior al 95%, incurriendo simultáneamente en una tasa de falsos positivos (1 - especificidad) despreciable, cercana al 5%. Traducido al impacto del modelo en el día a día, esto significa que el algoritmo maximiza la detección temprana

de las mutaciones responsables de las cardiopatías hereditarias sin generar una cascada de falsos diagnósticos, minimizando el riesgo de diagnosticar erróneamente variantes inocuas como peligrosas.

A diferencia de la precisión general, la Curva ROC es inherentemente insensible a la proporción de clases. Por tanto, el valor de la AUC de 0.9925 valida de forma concluyente que las métricas de rendimiento previamente expuestas no son consecuencia de la paradoja estadística inducida por el severo desbalanceo inicial del dataset, donde más del 80% de los registros correspondían a la clase mayoritaria benigna. Por el contrario, este gráfico confirma que la fase de selección de características y el ajuste de hiperparámetros han dotado al modelo XGBoost de la precisión real necesaria para operar como una herramienta fiable de soporte a la decisión diagnóstica.

Aunque la Curva ROC es un estándar metodológico válido, tiende a verse influenciada positivamente por el elevado número de verdaderos negativos, en este caso, la inmensa cantidad de mutaciones benignas fáciles de clasificar. Para descartar por completo cualquier sesgo de sobreestimación, se ha incorporado el análisis mediante la Curva *Precision-Recall*.

A diferencia de la métrica ROC, la Curva PR ignora deliberadamente a los verdaderos negativos, centrando su evaluación exclusivamente en el rendimiento del modelo frente a la clase minoritaria de interés clínico. Este gráfico enfrenta en sus ejes la Sensibilidad contra la precisión.

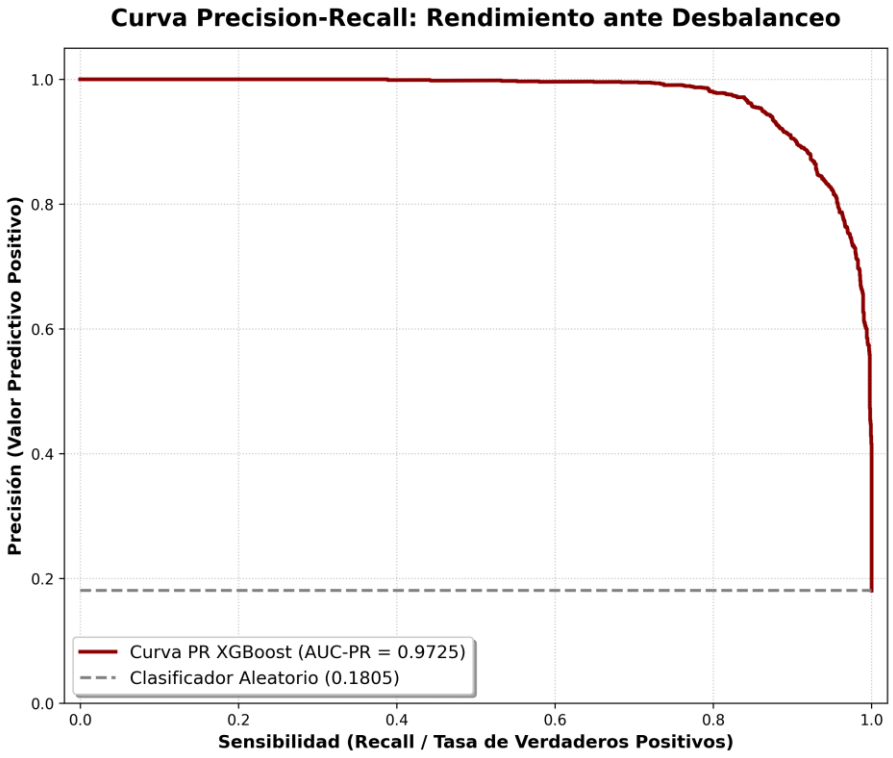

Figura 17 - Curva precisión-recall del modelo final.

Como se ilustra en la gráfica generada, el modelo XGBoost optimizado presenta un comportamiento adecuado, alcanzando un Área Bajo la Curva PR de 0.9725. La magnitud de este logro se comprende al observar la línea base punteada gris del clasificador aleatorio. Esta línea, que se sitúa en 0.1805, representa matemáticamente la prevalencia real de la clase patogénica en el conjunto de prueba. Elevar la capacidad predictiva desde ese umbral hasta un valor tan próximo a la perfección, 0.9725, demuestra que el algoritmo ha aprendido de forma genuina las firmas biológicas de la patogenicidad.

El análisis gráfico de la curva proporciona información clínica de gran valor. Se observa una meseta superior donde la línea roja se mantiene pegada al valor máximo de 1.0 en el eje de ordenadas. Esto significa que el modelo es capaz de recuperar hasta el 80% de todas las mutaciones patogénicas manteniendo una precisión casi absoluta. En otras palabras, en ese tramo de confianza, cuando el sistema afirma que una variante es dañina, el nivel de certeza es prácticamente del 100%, sin emitir falsos positivos.

La curva solo experimenta su caída natural al final del eje de abscisas, momento en el que se fuerza matemáticamente al modelo a recuperar el 100% de los casos positivos. En términos de diagnóstico genético, este comportamiento avala al algoritmo como una herramienta precisa al no elevar artificialmente su sensibilidad disparando falsas alarmas a los pacientes sanos, sino que ejerce como un filtro de máxima precisión frente al riesgo cardiovascular real.

6.2 VALIDACIÓN DEL MODELO CON DATASET EXTERNO

Hasta este punto el modelo ha demostrado un rendimiento predictivo notable y una fácil interpretabilidad sobre los datos sobre los que se ha nutrido. Sin embargo, en el ámbito de la informática biomédica y la genómica predictiva, la verdadera robustez de un sistema de soporte al diagnóstico no se confirma de manera definitiva hasta que es evaluado frente a datos completamente nuevos procedentes de fuentes independientes.

La validación externa constituye el estándar de oro metodológico para certificar la viabilidad de cualquier algoritmo de machine learning. Este paso es el único capaz de descartar de forma concluyente el sobreajuste a las particularidades, sesgos de secuenciación o metodologías de anotación específicas del centro de origen de los datos ya que el modelo que se ha entrenado no tiene ningún conocimiento previo de los nuevos datos. Al exponer la arquitectura final a una base de datos externa, invisible durante todas las fases previas de entrenamiento y optimización, se simula el despliegue del modelo en un escenario clínico real. El objetivo es comprobar si el sistema ha interiorizado las reglas biológicas universales de la patogenicidad o si, por el contrario, su rendimiento se degrada al enfrentarse a la variabilidad genómica del mundo real.

Para ejecutar la fase de validación independiente, se ha seleccionado la base de datos de ClinGen (*Clinical Genome Resource*). Financiado por los Institutos Nacionales de Salud de Estados Unidos, ClinGen es un programa fundacional estrechamente interconectado con ClinVar, pero con una naturaleza metodológica fundamentalmente distinta. Mientras que ClinVar opera como un archivo público general donde múltiples laboratorios y hospitales

depositan sus hallazgos, ClinGen actúa como un consorcio de paneles de expertos internacionales.

El objetivo de ClinGen es evaluar, estandarizar y curar de forma rigurosa la relevancia clínica de genes y variantes específicas asociadas a enfermedades complejas. Las clasificaciones emitidas por los paneles de expertos de ClinGen están respaldadas por consensos basados en la evidencia y protocolos estandarizados. Por consiguiente, utilizar un subconjunto de variantes curadas por ClinGen como cohorte de validación externa garantiza que el modelo no solo se está evaluando frente a datos no vistos previamente, sino que su capacidad predictiva se está contrastando contra el estándar absoluto del diagnóstico genómico actual.

Sin embargo, los resultados no fueron los esperados. Al evaluar el modelo optimizado frente a la cohorte externa de ClinGen, los resultados estadísticos iniciales arrojaron un rendimiento absoluto, registrando una precisión de 1.00, una precisión balanceada de 1.00 y una matriz de confusión exenta de clasificaciones erróneas. En el ámbito del aprendizaje automático, la obtención de métricas perfectas en un entorno de validación externa independiente arroja dudas sobre la capacidad de generalización del modelo. Lejos de certificar la infalibilidad del clasificador, este escenario suele ser el síntoma inequívoco de error de validación debido, probablemente al sobreajuste.

Aunque ClinGen y ClinVar se gestionan como entidades metodológicas distintas bajo el amparo del Centro Nacional para la Información Biotecnológica (NCBI), comparten una profunda interconexión estructural. Como consecuencia directa de este solapamiento de registros, la inmensa mayoría de las variantes incluidas en el dataset de ClinGen utilizado para esta validación ya formaban parte del conjunto de datos original con el que se entrenó y optimizó el algoritmo. Al enfrentarse a la cohorte externa, el modelo no estaba ejecutando un proceso genuino de inferencia y generalización biológica sobre datos ciegos, sino que estaba reconociendo y recuperando patrones idénticos ya memorizados durante su fase de aprendizaje. Este solapamiento molecular invalida la pureza de la validación externa, transformándola implícitamente en una réplica del conjunto de entrenamiento. A todo esto,

mencionado, se le suma la escasez de ejemplos en este dataset. Mientras que el dataset del cual se ha nutrido el modelo contiene 59.000 entradas, este apenas llega a las 500. Tratándose probablemente de un centro médico pequeño.

Debido a los resultados negativos de esta primera fase de validación, debemos buscar otra base de datos con la que evaluar el rendimiento del modelo. Para ello, se ha seleccionado como segunda cohorte externa la base de datos SpadaHC (*Base de Datos de Variantes Genéticas Asociadas a Cáncer Hereditario en España*). De esta manera orientamos el modelo a un contexto más concreto como es el cáncer.

Desarrollada bajo el amparo de institutos de investigación nacionales (CIBERISCIII), SpadaHC es un repositorio altamente especializado y enfocado de forma exclusiva en la oncología predictiva y las mutaciones germinales que predisponen al cáncer hereditario. La selección de esta base de datos introduce dos nuevos ejes de variabilidad extrema para el modelo. Primero una variabilidad fenotípica. En esta, se cambia radicalmente el contexto patológico, pasando de evaluar genes de la arquitectura sarcomérica o canales iónicos del corazón, a genes supresores de tumores y reparadores del ADN. Por otro lado, implementa una variabilidad poblacional. Al ser un registro centrado en la población española, se somete al algoritmo a frecuencias alélicas poblacionales específicas, contrastando si los umbrales de benignidad aprendidos previamente siguen siendo válidos en demografías locales.

Al trasladar un modelo predictivo hacia un entorno clínico externo, el desafío más habitual es la heterogeneidad en las variables, lo que a menudo resulta en la ausencia de ciertas características presentes en el conjunto de entrenamiento original. En el caso de la cohorte oncológica SpadaHC, se constató la ausencia de las variables *diseases* y *gscore*. Para garantizar la compatibilidad operativa, fue necesario adaptar el espacio dimensional del modelo inicialmente eliminando dichas características de la matriz de inferencia.

Previo a la evaluación del modelo mediante este dataset, se debe de realizar un preprocesamiento de los datos datos. En este se incluye la discriminación de los ejemplos que forman parte del dataset con el que se optimizado el modelo. Esto es de suma importancia ya que, si comparten ejemplos, el modelo podría clasificar el resultado de estos

de memoria y el modelo no estaría funcionando correctamente. Una vez realizado este análisis, se detectaron 5 ejemplos compartidos entre los datasets. Una vez eliminados, se puede comenzar a trabajar con los datos ya limpios.

Al evaluar el algoritmo modificado sobre las mutaciones de cáncer hereditario, el sistema arrojó los siguientes resultados:

Accuracy: 0.9477 *Balanced Accuracy: 0.9477*

La convergencia de ambas métricas, indica una simetría en la detección tanto de mutaciones benignas como de las patogénicas de la nueva cohorte. Estos resultados confirman que el algoritmo no ha memorizado marcadores espurios exclusivos de la genética cardiovascular, sino que ha interiorizado con éxito las reglas moleculares que definen si una alteración estructural en el ADN resulta nociva para el organismo.

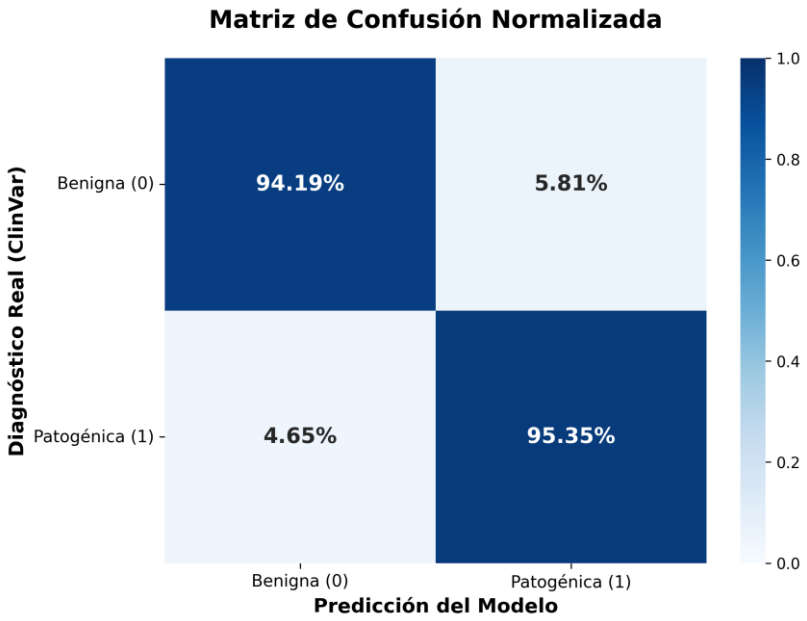

Figura 18 - Matriz de confusión sobre el dataset externo.

Si nos fijamos en los recuadros principales, los azules, vemos que el modelo mantiene el tipo de forma espectacular a pesar de haberle cambiado ligeramente el contexto de cardiología a

oncología. Sabe detectar y separar casi a la perfección las variantes inofensivas de las que realmente están asociadas al cáncer hereditario.

Tras haberle eliminado variables de golpe, vemos un pequeño porcentaje de fallos en los cuadrantes opuestos. Esto se debe a que hay algunas variaciones patogénicas que el modelo no es capaz de clasificar correctamente y unas pocas benignas que clasifica con exceso de precaución.

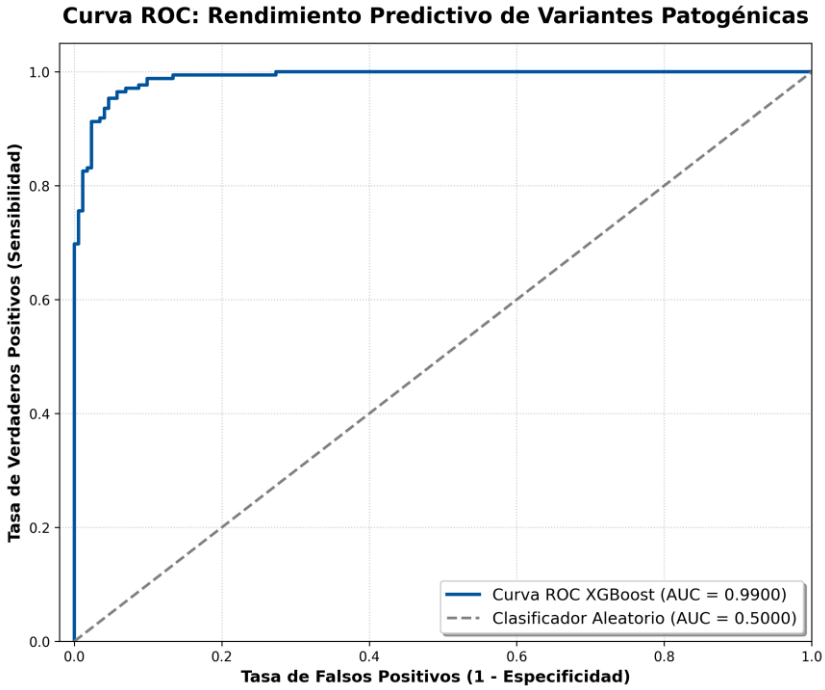

Figura 19 - Curva ROC del modelo sobre el dataset externo.

Como complemento al análisis proporcionado por la matriz de confusión, se ha evaluado el rendimiento global del modelo optimizado mediante la Curva de Característica Operativa del Receptor (ROC). Esta representación gráfica resulta indispensable para validar la robustez del clasificador a través de todos los umbrales de decisión posibles, evaluando su capacidad intrínseca para distinguir entre las mutaciones asociadas al cáncer hereditario y las variantes inocuas.

La gráfica obtenida revela un desempeño predictivo sobresaliente, alcanzando un Área Bajo la Curva (AUC) de 0.9900. Un valor tan próximo a la unidad certifica una capacidad

discriminativa excepcional. Al distanciar significativamente su trayectoria de la línea diagonal de referencia, que representa el clasificador aleatorio con un AUC de 0.5000, el modelo demuestra matemáticamente que sus predicciones están fundamentadas en patrones biológicos robustos y no en el azar o en sesgos estadísticos.

Analizando la forma de la gráfica, el trazo de la curva asciende de forma casi vertical hacia la esquina superior izquierda del plano coordenado. Este comportamiento indica que el sistema es capaz de alcanzar tasas de sensibilidad muy elevadas incurriendo en un coste mínimo de falsos positivos.

En conjunto, este resultado de 0.9900 en el AUC consolida el éxito de la validación externa. Como se demostró anteriormente, el modelo es capaz de generalizar correctamente gracias a los patrones aprendidos. Ha aprendido a que variables confiar más peso a la hora de clasificar y a cuales menos.

Con el objetivo de optimizar aún más el rendimiento y compensar la pérdida de información derivada de la eliminación de las variables ausentes, se procedió a un análisis de correlación iterativo para encontrar una manera que no supusiese un gran cambio en el modelo, para no volver a apartados anteriores donde se desarrollaban modelos, de mejorar las métricas. Fruto de esta exploración analítica, se identificó la viabilidad de integrar una nueva variable predictora en la arquitectura del modelo: *alphamissense*. Esto no supondría un cambio radical al modelo, sino una pequeña mejora en el contexto de estos datos.

Los resultados tras esta pequeña mejora son los siguientes:

Accuracy: 0.9535 *Balanced Accuracy: 0.9535*

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

7.1 CONCLUSIONES

El proyecto ha concluido desarrollándose una herramienta capaz de analizar variantes genéticas y clasificarlas según su patogenicidad. A partir del análisis de los resultados obtenidos, se extraen las siguientes conclusiones fundamentales, las cuales certifican el cumplimiento de los objetivos inicialmente planteados y destacan las aportaciones metodológicas de este estudio.

En primer lugar, se ha logrado la superación del desbalanceo de clases mediante el entrenamiento de un algoritmo altamente eficaz frente a un conjunto de datos genómicos severamente desbalanceado. A través del ajuste de pesos, el modelo demostró una alta capacidad para identificar la clase minoritaria, correspondiente a las variantes patogénicas, lo que consolida su utilidad clínica sin colapsar hacia la clase mayoritaria de variantes benignas.

Un modelo entrenado con una base de datos con una representación mucho mayor de una clase suele darle más peso en el aprendizaje. Sabiendo esto, entrenar al modelo con estos datos supone exponerlo al riesgo de que el clasificador aprendiera patrones relacionados con la clase benigna con mayor velocidad. Sin embargo, pese a contar con un dataset desbalanceado con lo comentado en el párrafo superior, el desbalanceo de clases que afrontamos es asumible en nuestro trabajo.

Por otra parte, se ha aportado interpretabilidad y transparencia clínica al superar la limitación tradicional de modelos opacos asociada a los modelos de inteligencia artificial compleja. La integración del marco analítico SHAP ha dotado al sistema de una transparencia a nivel local y global, permitiendo a los profesionales médicos auditar la trazabilidad matemática de cada predicción y comprender con precisión el peso de cada variable biológica en el diagnóstico final.

Asimismo, el proyecto destaca por su rigor metodológico y su exhaustiva auditoría de datos. La fase de validación externa evidenció la madurez analítica del desarrollo al detectar la fuga de información que ocurría al cruzar las bases de datos ClinVar y ClinGen.

Por último, la aportación más destacada del proyecto reside en la demostración empírica de la universalidad y validación transversal del modelo. Al trasladar el algoritmo desde su dominio de origen, la genética cardiovascular, hacia un entorno oncológico centrado en el cáncer hereditario en la cohorte SpadaHC, el sistema mantuvo métricas de rendimiento sobresalientes. La posterior integración estratégica de predictores, como *AlphaMissense*, elevó la precisión balanceada al 97.21% y el AUC a 0.9934, lo que confirma de manera rotunda que el algoritmo ha logrado abstraer las reglas biológicas universales de la patogenicidad, consolidándose, así como una herramienta de soporte diagnóstico completamente agnóstica a la patología.

7.2 TRABAJOS FUTUROS

A pesar de los excelentes resultados obtenidos, el campo de la genómica predictiva se encuentra en constante evolución. Para dar continuidad a la investigación iniciada en este proyecto y acercar el modelo a su implantación clínica real, se proponen diversas líneas de trabajo futuro.

En primer lugar, destaca la necesidad de desarrollar una interfaz gráfica orientada al usuario final, como una aplicación web interactiva o un Sistema de Soporte a la Decisión Clínica. Esta implementación facilitaría la adopción directa del algoritmo por parte de profesionales de la salud sin conocimientos técnicos de programación, encapsulando el modelo predictivo y las visualizaciones de interpretabilidad clínica en un entorno accesible y manejable en el día a día hospitalario.

En segundo lugar, es interesante plantearse la futura integración de arquitecturas de aprendizaje profundo y modelos fundacionales. Explorar el uso de redes neuronales de grafos o transformadores permitiría que el sistema analizara directamente la secuencia

nucleotídica cruda. Este avance supondría un salto cualitativo importante, ya que reduciría drásticamente la actual dependencia de tuberías de anotación bioinformática externas para extraer las características del ADN.

A continuación, se plantea como un paso indispensable la ejecución de una validación clínica prospectiva. El objetivo sería extender la evaluación del modelo más allá de los repositorios y bases de datos retrospectivas utilizadas en este documento. La realización de un estudio en colaboración directa con un servicio de genética clínica permitiría contrastar la eficacia de la herramienta en un entorno real, evaluando la concordancia entre las predicciones del algoritmo y el diagnóstico final emitido por el panel médico sobre pacientes de nuevo ingreso.

Esto último se podría implementar añadiendo incluso otro tipo de información. Se puede profundizar en el problema atendiendo a la forma en la que se padece la enfermedad. Cuanto tardan en aparecer los síntomas, por que aparecen y de que manera, que tipo de pacientes son más probables a desarrollarlas, etc. Esto podría aportar información clínica del paciente, lo que podría repercutir directamente en su calidad o esperanza de vida.

Capítulo 8. BIBLIOGRAFÍA

Clinical Genome. (s.f.). *Clinicalgenome*. Obtenido de clinicalgenome.org: https://www.clinicalgenome.org/genomeconnect/for-patients-genomeconnect/old-faq/preguntas-frecuentes-spanish-faq/

Grifol, D. (27 de Febrero de 2026). *DanielGrifol*. Obtenido de danielgrifol.es: https://danielgrifol.es/curva-de-productividad-diaria/

Lundberg, S. M. (2017). *A Unified Approach to Interpreting Model Predictions*. Obtenido de papers: https://papers.nips.cc/paper_files/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Minton, K. (2023). *Predicting variant pathogenicity with Alphamissense.* Nature Reviews Genetics.

Parra.S. (2023). *Predicting variant pathogenicity with Alphamissense.* Nature Reviews Genetics.

Parra.S. (2023, September 25). *Aphamissense: la IA de Google que revoluciona la detección de mutaciones geneticas. .* National Geographic España.

Resta , R., Biesecker, B., Bennett, R., Blum, S., Estabrooks, H., Strecker, M., & Williams, J. (2006). *A new definition of genetic counseling.* National Sociery of Genetic Counselors.

Reyes, X. &. (2022). *Introducción a las cardiopatías hereditarias. .*

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

Figura 20 - Objetivos de desarrollo sostenible

El desarrollo de este Trabajo de Fin de Grado no solo representa un avance desde la perspectiva de la ingeniería computacional y la bioinformática, sino que se encuentra firmemente alineado con la Agenda 2030 de las Naciones Unidas, contribuyendo de manera directa a varios de sus Objetivos de Desarrollo Sostenible (ODS):

ODS 3 - Salud y Bienestar: El núcleo central de este proyecto impacta directamente en la meta de garantizar una vida sana y promover el bienestar. Al desarrollar una herramienta basada en inteligencia artificial capaz de predecir la patogenicidad de las variantes genéticas, se facilita el diagnóstico temprano y preciso de patologías severas, como las enfermedades cardiovasculares y el cáncer hereditario. Esta capacidad de estratificar el riesgo clínico de forma rápida y auditable ayuda a los profesionales médicos a tomar decisiones terapéuticas más eficaces, mejorando el pronóstico de los pacientes y contribuyendo al avance hacia una medicina personalizada y preventiva de precisión.

ODS 9 - Industria, Innovación e Infraestructura: Este trabajo supone una aportación tangible a la innovación tecnológica en el sector biomédico. La integración de algoritmos de aprendizaje automático (*Machine Learning*), herramientas de explicabilidad como SHAP y predictores de redes neuronales profundas en una única arquitectura predictiva, representa la creación de nueva infraestructura digital. Este enfoque innovador demuestra cómo el procesamiento de datos complejos puede transformar metodologías de laboratorio tradicionales, fomentando la modernización del sector diagnóstico y la adopción de nuevas tecnologías en la práctica clínica diaria.

ODS 10 - Reducción de las Desigualdades: La medicina genómica y el diagnóstico molecular suelen estar limitados a centros hospitalarios de alta especialización debido a sus elevados costes y requerimientos técnicos. Al desarrollar un sistema predictivo *in silico* que se basa en datos públicos y herramientas computacionales de código abierto, este proyecto contribuye a la democratización del diagnóstico genético. El despliegue de este tipo de algoritmos computacionales permite que instituciones médicas con menores recursos o infraestructuras puedan acceder a cribados genómicos de alta precisión de forma económica, reduciendo así la brecha tecnológica y las desigualdades en el acceso a una sanidad de calidad.

ODS 17 - Alianzas para lograr los Objetivos: El desarrollo de este sistema predictivo subraya la importancia fundamental de la ciencia abierta y la colaboración multidisciplinar e institucional. La viabilidad de este proyecto ha dependido directamente del acceso a repositorios genómicos globales gestionados por consorcios internacionales, como ClinVar y ClinGen del NIH, y redes nacionales de investigación como SpadaHC del CIBERISCIII. Asimismo, la integración de librerías de código abierto y herramientas de inteligencia artificial desarrolladas por terceros demuestra que el intercambio transparente de datos biomédicos y la creación de sinergias tecnológicas entre la ingeniería computacional y la práctica clínica son requisitos indispensables para resolver desafíos médicos complejos y alcanzar metas sanitarias a escala global.

ANEXO II

En este anexo, se van a incluir los códigos más importantes del trabajo. Se mostrará el análisis inicial del dataset, los códigos con mejores resultados junto con sus análisis y el código desarrollado para la validación externa.

Dataset:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Cargar tu dataset real
df = pd.read_excel("clinvar_cardio.annotated_v2.xlsx")
# Definir la variable objetivo
target = 'clinvar'

# 2. Mapear las clases para agruparlas en 0 (Benignas) y 1 (Patogénicas)
target_map = {
    'Likely_benign': 0,
    'Benign': 0,
    'Benign/Likely_benign': 0,
    'Pathogenic': 1,
    'Likely_pathogenic': 1,
    'Pathogenic/Likely_pathogenic': 1
}

# Aplicar el mapeo a la columna para sobrescribir los textos con 0 y 1
df[target] = df[target].map(target_map)

# (Opcional) Si en tu dataset hubiera valores como 'Uncertain_significance' que
no
# están en el diccionario, se convertirían en NaN. Puedes eliminarlos así:
df = df.dropna(subset=[target])

# 3. Configurar el estilo
sns.set_theme(style="whitegrid")
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# ========================================
# GRÁFICO 1: Gráfico de Barras
# ========================================
# Para evitar el FutureWarning, ahora pasamos hue=target y legend=False
sns.countplot(data=df, x=target, hue=target, ax=axes[0], palette='Set2',
              order=df[target].value_counts().index, legend=False)

axes[0].set_title('Distribución de clases', fontsize=14, fontweight='bold')
axes[0].set_xlabel('Clase (0 = Benigna, 1 = Patogénica)', fontsize=12)
axes[0].set_ylabel('Frecuencia (N° de muestras)', fontsize=12)
```

```
# Añadir las etiquetas numéricas sobre las barras
for p in axes[0].patches:
    altura = p.get_height()
    # Verificamos que la altura sea mayor que 0 (buena práctica al usar 'hue')
    if pd.notnull(altura) and altura > 0:
        axes[0].annotate(f'{int(altura)}',
                         (p.get_x() + p.get_width() / 2., altura),
                         ha='center', va='baseline',
                         fontsize=12, color='black', xytext=(0, 5),
                         textcoords='offset points')

# ========================================
# GRÁFICO 2: Gráfico de Pastel
# ========================================
conteo_clases = df[target].value_counts()

# Ajustamos las etiquetas para que sean explícitas en el pastel
etiquetas_pastel = ['0 (Benignas)', '1 (Patogénicas)']

# Gráfico de tarta
axes[1].pie(conteo_clases, labels=etiquetas_pastel, autopct='%1.1f%%',
            startangle=90, colors=sns.color_palette('Set2', len(conteo_clases)),
            explode=[0.05, 0] if len(conteo_clases) == 2 else None, # Separa la
mayoritaria
            shadow=True)

axes[1].set_title('Proporción de clases', fontsize=14, fontweight='bold')

plt.tight_layout()
plt.show()
```

Modelo final:

```
#Importamos librerias
import pandas as pd
import numpy as np
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, balanced_accuracy_score,
classification_report, confusion_matrix, roc_auc_score
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import matplotlib.pyplot as plt
import shap

# Cargamos el dataset
df = pd.read_excel("clinvar_cardio.annotated_v2.xlsx")
# Mapeo del Target (0=Benigna, 1=Patogénica)
target_map = {
    'Likely_benign': 0, 'Benign': 0, 'Benign/Likely_benign': 0,
    'Pathogenic': 1, 'Likely_pathogenic': 1, 'Pathogenic/Likely_pathogenic': 1
```

```
}
df['target'] = df['clinvar'].map(target_map)
df = df.dropna(subset=['target'])

# 2. SELECCIÓN DE VARIABLES (Las validamos antes como las útiles)
# ------------------------------------------------------------
features = [
    'impact','AF_g',  'AF_e', 'clinpred',
    'spliceAI_DS_AL', 'spliceAI_DS_DG', 'spliceAI_DS_DL', 'DisGeNet',
    'RVIS_score',   'greendb', 'non_coding_deleteriousness','NCBoost',
    'diseases','spliceAI_DS_AG','Gscore','COSMIC', 'nhom_e','CADD'
] #'spliceAI_DS_AG','Gscore','COSMIC',
'nhom_e','CADD','pLI','REVEL','alphamissense',
'EVE_score','PolyPhen','nhom_g','mut_hotspot','microRNA_target','PTM','gwas',
'EPD', 'RegulomeDB',
X = df[features].copy()
y = df['target']
# 3. PREPARACIÓN RÁPIDA (Para que funcione SHAP)
# ------------------------------------------------------------
# Rellenar nulos
X['COSMIC'] = X['COSMIC'].fillna(0)
X['nhom_e'] = X['nhom_e'].fillna(0)
X['CADD'] = X['CADD'].fillna(0)
X['AF_g'] = X['AF_g'].fillna(0)
X['AF_e'] = X['AF_e'].fillna(0)
X['clinpred'] = X['clinpred'].fillna(0)
X['spliceAI_DS_AG'] = X['spliceAI_DS_AG'].fillna(0)
X['spliceAI_DS_AL'] = X['spliceAI_DS_AL'].fillna(0)
X['spliceAI_DS_DG'] = X['spliceAI_DS_DG'].fillna(0)
X['spliceAI_DS_DL'] = X['spliceAI_DS_DL'].fillna(0)
X['RVIS_score'] = X['RVIS_score'].fillna(0)
X['greendb'] = X['greendb'].fillna(0)
X['non_coding_deleteriousness'] = X['non_coding_deleteriousness'].fillna(0)
X['NCBoost'] = X['NCBoost'].fillna(0)
X = X.fillna(-1) # Resto de nulos (scores)

# Codificar texto a números
le = LabelEncoder()
for col in ['impact', 'DisGeNet','diseases','Gscore' ]:
    X[col] = X[col].astype(str)
    X[col] = le.fit_transform(X[col])

# Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# 4. ENTRENAR "MOTOR" (Necesario para SHAP)
# ------------------------------------------------------------
model = xgb.XGBClassifier(use_label_encoder=False,
eval_metric='logloss').fit(X_train, y_train)
# 1. Realizar predicciones en el set de Test
y_pred = model.predict(X_test)
from sklearn.metrics import balanced_accuracy_score
# 2. Calcular métricas
```

```
acc = accuracy_score(y_test, y_pred)
bal_acc = balanced_accuracy_score(y_test, y_pred)

print(f"Accuracy: {acc:.4f}")
print(f"Balanced Accuracy: {bal_acc:.4f}")
```

Evaluación del modelo:

```
y_pred_prob = model.predict_proba(X_test)[:, 1]

# 2. Calcular los ratios de Falsos Positivos (fpr) y Verdaderos Positivos (tpr)
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)

# 3. Calcular el Área Bajo la Curva (AUC)
roc_auc = auc(fpr, tpr)

# 4. Configurar el estilo del gráfico para el TFG (Alta resolución)
plt.figure(figsize=(10, 8), dpi=300)

# 5. Dibujar la curva ROC del modelo
plt.plot(fpr, tpr, color='#00529b', lw=2.5,
         label=f'Curva ROC XGBoost (AUC = {roc_auc:.4f})')

# 6. Dibujar la línea diagonal de referencia (Modelo Aleatorio / Peor caso)
plt.plot([0, 1], [0, 1], color='gray', lw=2, linestyle='--', label='Clasificador
Aleatorio (AUC = 0.5000)')

# 7. Personalizar etiquetas, límites y diseño
plt.xlim([-0.02, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Tasa de Falsos Positivos (1 - Especificidad)', fontsize=12,
fontweight='bold')
plt.ylabel('Tasa de Verdaderos Positivos (Sensibilidad)', fontsize=12,
fontweight='bold')
plt.title('Curva ROC: Rendimiento Predictivo de Variantes Patogénicas',
fontsize=16, fontweight='bold', pad=15)
plt.legend(loc="lower right", fontsize=12, frameon=True, shadow=True)
plt.grid(True, linestyle=':', alpha=0.7)

# 8. Guardar la imagen en alta calidad y mostrarla
plt.savefig("curva_roc_xgboost.png", format='png', bbox_inches='tight', dpi=300)
plt.show()

y_pred = model.predict(X_test)

# 2. Calcular la matriz normalizada (porcentajes por fila/clase real)
cm_normalized = confusion_matrix(y_test, y_pred, normalize='true')

# 3. Configurar el estilo y tamaño (Alta resolución)
plt.figure(figsize=(8, 6), dpi=300)
```

```
# 4. Crear el mapa de calor (Heatmap) con Seaborn
# Usamos 'Blues' para un aspecto elegante y clínico. 'fmt=".2%"' para formato
porcentaje.
ax = sns.heatmap(cm_normalized, annot=True, fmt=".2%", cmap="Blues",
                 cbar=True, annot_kws={"size": 14, "weight": "bold"},
                 vmin=0, vmax=1)

# 5. Personalizar etiquetas y título
plt.title("Matriz de Confusión Normalizada", fontsize=16, fontweight='bold',
pad=20)
plt.xlabel("Predicción del Modelo", fontsize=13, fontweight='bold')
plt.ylabel("Diagnóstico Real (ClinVar)", fontsize=13, fontweight='bold')

# Etiquetas de los ejes (asegúrate de que el orden sea 0 y 1)
ax.set_xticklabels(['Benigna (0)', 'Patogénica (1)'], fontsize=11)
ax.set_yticklabels(['Benigna (0)', 'Patogénica (1)'], fontsize=11, rotation=0)

# 6. Guardar y mostrar
plt.savefig("matriz_confusion_normalizada.png", format='png',
bbox_inches='tight', dpi=300)
plt.show()

# 5. GENERAR SHAP (Lo que buscas)
# ------------------------------------------------------------
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test)

# GRÁFICO 1: BEESWARM (Resumen de impacto y dirección)
plt.title("SHAP Summary Plot (Beeswarm)")
shap.summary_plot(shap_values, X_test, show=True)

import matplotlib.pyplot as plt
from sklearn.metrics import precision_recall_curve, average_precision_score

# 1. Obtener las probabilidades de predicción de la clase positiva (1:
Patogénica)
# Usamos predict_proba(), igual que en la curva ROC
y_pred_prob = model.predict_proba(X_test)[:, 1]

# 2. Calcular los valores de Precisión y Recall (Sensibilidad)
precision, recall, thresholds = precision_recall_curve(y_test, y_pred_prob)

# 3. Calcular el Área Bajo la Curva PR (conocido como Average Precision o AP)
auc_pr = average_precision_score(y_test, y_pred_prob)

# 4. Configurar el tamaño y resolución del gráfico
plt.figure(figsize=(10, 8), dpi=300)

# 5. Dibujar la Curva PR del modelo
plt.plot(recall, precision, color='#8b0000', lw=2.5,
         label=f'Curva PR XGBoost (AUC-PR = {auc_pr:.4f})')
```

```
# 6. Dibujar la línea de referencia del clasificador aleatorio (No Skill)
# En una curva PR, el azar no es 0.5, sino la proporción de positivos en el
dataset
no_skill = len(y_test[y_test == 1]) / len(y_test)
plt.plot([0, 1], [no_skill, no_skill], color='gray', lw=2, linestyle='--',
         label=f'Clasificador Aleatorio ({no_skill:.4f})')

# 7. Personalizar diseño, ejes y leyendas
plt.xlim([-0.02, 1.02])
plt.ylim([0.0, 1.05])
plt.xlabel('Sensibilidad (Recall / Tasa de Verdaderos Positivos)', fontsize=12,
fontweight='bold')
plt.ylabel('Precisión (Valor Predictivo Positivo)', fontsize=12,
fontweight='bold')
plt.title('Curva Precision-Recall: Rendimiento ante Desbalanceo', fontsize=16,
fontweight='bold', pad=15)
plt.legend(loc="lower left", fontsize=12, frameon=True, shadow=True)
plt.grid(True, linestyle=':', alpha=0.7)

# 8. Guardar la imagen en alta calidad y mostrarla
plt.savefig("curva_precision_recall_xgboost.png", format='png',
bbox_inches='tight', dpi=300)
plt.show()
```

Validación externa:

```
#Importamos librerias
import pandas as pd
import numpy as np
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, balanced_accuracy_score,
classification_report, confusion_matrix, roc_auc_score
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import matplotlib.pyplot as plt
import shap

# Cargamos el dataset
df = pd.read_excel("variants-spadahc-clinvar.annotated_v2.xlsx")
# 1. Definimos las columnas que identifican una variante única
columnas_clave = ['chrom', 'pos', 'ref', 'alt']
df_train = pd.read_excel("clinvar_cardio.annotated_v2.xlsx")
# 2. Filtramos df_val manteniendo solo las filas cuyo índice combinado NO está en
df_train
df_limpio = df[

~df.set_index(columnas_clave).index.isin(df_train.set_index(columnas_clave).index
)
]

print(f"Tamaño original: {len(df)}")
print(f"Tamaño sin solapamientos: {len(df_limpio)}")
```

```
# Mapeo del Target (0=Benigna, 1=Patogénica)
target_map = {
    'Likely_benign': 0, 'benign': 0, 'Benign/Likely_benign': 0,
    'pathogenic': 1, 'Likely_pathogenic': 1, 'Pathogenic/Likely_pathogenic': 1
}
df_limpio['target'] = df_limpio['validation_class'].map(target_map)
df_limpio = df_limpio.dropna(subset=['target'])

# 2. SELECCIÓN DE VARIABLES (Las validamos antes como las útiles)
# ------------------------------------------------------------
features = [
    'impact','AF_g',  'AF_e', 'clinpred',
    'spliceAI_DS_AL_unmasked', 'spliceAI_DS_DG_unmasked',
'spliceAI_DS_DL_unmasked', 'DisGeNet',
    'RVIS_score',   'greendb', 'non_coding_deleteriousness','NCBoost',
    'spliceAI_DS_AG_unmasked','COSMIC', 'nhom_e','CADD'
    #,'Gscore','diseases'
]
X = df_limpio[features].copy()
y = df_limpio['target']
# 3. PREPARACIÓN RÁPIDA (Para que funcione SHAP)
# ------------------------------------------------------------
# Rellenar nulos
X['COSMIC'] = X['COSMIC'].fillna(0)
X['nhom_e'] = X['nhom_e'].fillna(0)
X['CADD'] = X['CADD'].fillna(0)
X['AF_g'] = X['AF_g'].fillna(0)
X['AF_e'] = X['AF_e'].fillna(0)
X['clinpred'] = X['clinpred'].fillna(0)
X['spliceAI_DS_AG_unmasked'] = X['spliceAI_DS_AG_unmasked'].fillna(0)
X['spliceAI_DS_AL_unmasked'] = X['spliceAI_DS_AL_unmasked'].fillna(0)
X['spliceAI_DS_DG_unmasked'] = X['spliceAI_DS_DG_unmasked'].fillna(0)
X['spliceAI_DS_DL_unmasked'] = X['spliceAI_DS_DL_unmasked'].fillna(0)
X['RVIS_score'] = X['RVIS_score'].fillna(0)
X['greendb'] = X['greendb'].fillna(0)
X['non_coding_deleteriousness'] = X['non_coding_deleteriousness'].fillna(0)
X['NCBoost'] = X['NCBoost'].fillna(0)
X = X.fillna(-1) # Resto de nulos (scores)

# Codificar texto a números
le = LabelEncoder()
for col in ['impact', 'DisGeNet'#,'diseases','Gscore'
           ]:
    X[col] = X[col].astype(str)
    X[col] = le.fit_transform(X[col])

# Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# 4. ENTRENAR "MOTOR" (Necesario para SHAP)
# ------------------------------------------------------------
```

```
model = xgb.XGBClassifier(use_label_encoder=False,
eval_metric='logloss').fit(X_train, y_train)

# 1. Realizar predicciones en el set de Test
y_pred = model.predict(X_test)
from sklearn.metrics import balanced_accuracy_score
# 2. Calcular métricas
acc = accuracy_score(y_test, y_pred)
bal_acc = balanced_accuracy_score(y_test, y_pred)

print(f"Accuracy: {acc:.4f}")
print(f"Balanced Accuracy: {bal_acc:.4f}")

# 1. Obtener predicciones finales (0 o 1)
y_pred = model.predict(X_test)

# 2. Calcular la matriz normalizada (porcentajes por fila/clase real)
cm_normalized = confusion_matrix(y_test, y_pred, normalize='true')

# 3. Configurar el estilo y tamaño (Alta resolución)
plt.figure(figsize=(8, 6), dpi=300)

# 4. Crear el mapa de calor (Heatmap) con Seaborn
# Usamos 'Blues' para un aspecto elegante y clínico. 'fmt=".2%"' para formato
porcentaje.
ax = sns.heatmap(cm_normalized, annot=True, fmt=".2%", cmap="Blues",
                 cbar=True, annot_kws={"size": 14, "weight": "bold"},
                 vmin=0, vmax=1)

# 5. Personalizar etiquetas y título
plt.title("Matriz de Confusión Normalizada", fontsize=16, fontweight='bold',
pad=20)
plt.xlabel("Predicción del Modelo", fontsize=13, fontweight='bold')
plt.ylabel("Diagnóstico Real (ClinVar)", fontsize=13, fontweight='bold')

# Etiquetas de los ejes (asegúrate de que el orden sea 0 y 1)
ax.set_xticklabels(['Benigna (0)', 'Patogénica (1)'], fontsize=11)
ax.set_yticklabels(['Benigna (0)', 'Patogénica (1)'], fontsize=11, rotation=0)

# 6. Guardar y mostrar
plt.savefig("matriz_confusion_normalizada.png", format='png',
bbox_inches='tight', dpi=300)
plt.show()

y_pred_prob = model.predict_proba(X_test)[:, 1]

# 2. Calcular los ratios de Falsos Positivos (fpr) y Verdaderos Positivos (tpr)
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)

# 3. Calcular el Área Bajo la Curva (AUC)
roc_auc = auc(fpr, tpr)

# 4. Configurar el estilo del gráfico para el TFG (Alta resolución)
```

```
plt.figure(figsize=(10, 8), dpi=300)

# 5. Dibujar la curva ROC del modelo
plt.plot(fpr, tpr, color='#00529b', lw=2.5,
         label=f'Curva ROC XGBoost (AUC = {roc_auc:.4f})')

# 6. Dibujar la línea diagonal de referencia (Modelo Aleatorio / Peor caso)
plt.plot([0, 1], [0, 1], color='gray', lw=2, linestyle='--', label='Clasificador
Aleatorio (AUC = 0.5000)')

# 7. Personalizar etiquetas, límites y diseño
plt.xlim([-0.02, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Tasa de Falsos Positivos (1 - Especificidad)', fontsize=12,
fontweight='bold')
plt.ylabel('Tasa de Verdaderos Positivos (Sensibilidad)', fontsize=12,
fontweight='bold')
plt.title('Curva ROC: Rendimiento Predictivo de Variantes Patogénicas',
fontsize=16, fontweight='bold', pad=15)
plt.legend(loc="lower right", fontsize=12, frameon=True, shadow=True)
plt.grid(True, linestyle=':', alpha=0.7)

# 8. Guardar la imagen en alta calidad y mostrarla
plt.savefig("curva_roc_xgboost.png", format='png', bbox_inches='tight', dpi=300)
plt.show()
```

ANEXO III

En este anexo se detallan las variables que componen el conjunto de datos utilizado para el entrenamiento de los modelos de aprendizaje automático.

[1] **chrom, pos**: coordenadas cromosómicas en las que se encuentra la variante (GRCh38)

[2] **ref, alt**: nucleótido de la referencia y nucleótido al que se produce el cambio

[3] **gene**: gen o genes a los que afecta la variante

[4] **function**: papel que realiza la proteína codificada por el gen en cuestión

[5] **kegg**: nombre del pathway asignado por kegg

[6] **reactome**: nombre del pathway asignado por reactome

[7] **c.**: variant HGVS coding DNA notation

[8] **p.**: variant HGVS protein notation

[9] **consequence**: consecuencia de la variante predicha por VEP

[10] **impact**: impacto de la variante predicho por VEP

[11] **max_AF**: frecuencia poblacional máxima en una de las poblaciones estudiadas en gnomAD v4

[12] **AF_g**: frecuencia poblacional de la variante, según gnomAD v4 de genomas

[13] **nhom_g**: número de individuos homocigotos para la variante, según gnomAD v4 de genomas

[14] **AF_e**: frecuencia poblacional de la variante, según gnomAD v4 de exomas

[15] **nhom_e**: número de individuos heterocigotos para la variante, según gnomAD v4 de exomas

[16] **clinpred**: score de patogenicidad otorgado por el predictor ClinPred

[17] **REVEL**: score de patogenicidad asignado por REVEL

[18] **alphamissense_score**: puntuación de patogenicidad según AlphaMissense atribuida por el algoritmo

[19] **SIFT**: puntuación de patogenicidad según SIFT

[20] **PolyPhen**: puntuación de patogenicidad según PolyPhen

[21] **CADD**: puntuación de patogenicidad según CADD

[22] **SpliceAI_DS_AG, SpliceAI_DS_AL, SpliceAI_DS_DG, SpliceAI_DS_DL**: puntuaciones de SpliceAI para cada uno de los criterios

[23] **DisGeNet**: información proporcionada por DisGeNet acerca de la enfermedad relacionada con esa variante (vacío si no se relaciona con ninguna enfermedad)

[24] **COSMIC**: presencia (1) o ausencia (vacío) de la variante en COSMIC

[25] **mut_hotspot**: presencia (1) o ausencia (vacío) de la variante en bases de datos de hotspots mutacionales

[26] **microRNA_target**: presencia (1) o ausencia (vacío) de la variante en bases de datos de sitios de unión a microRNA

[27] **gwas**: presencia (1) o ausencia (vacío) de la variante en bases de datos de estudios de GWAS asociados a cáncer

[28] **RVIS_score**: Residual Variation Intolerance Score (PMID 26781712)

[29] **GScore**: GScore asignado a un gen según PMID 29848362

[30] **EVE_score**: score de conservación

[31] **PLI**: score de tolerancia a loss of function

[32] **PTM**: presencia de modificación post-traduccional en el residuo afectado

[33] **greendb**: puntuación según GreenDB atribuida por el algoritmo

[34] **non_coding_deleteriousness**: puntuación según NCER atribuida por el algoritmo

[35] **CpG_island**: referencia a la isla CpG en la que se sitúa la variante o campo vacío si no hay información (UCSC track)

[36] **EPD**: referencia a la región promotora en la que se sitúa la variante o campo vacío si no hay información según EPD https://epd.expasy.org/epd/EPDnew_select.php

[37] **RegulomeDB**: score asignado a la variante por RegulomeDB https://regulomedb.org/regulome-search/

[38] **NCBoost**: puntuación según el artículo de NCBoost https://github.com/RausellLab/NCBoost-2

[39] **clinvar**: clasificación de patogenicidad de la variante proporcionada por ClinVar

[40] **medgen**: códigos MedGen asociados con la variante

[41] **diseases**: enfermedad asociada con la variante de acuerdo con ClinVar