Predicción del desenlace terapéutico para Leishmaniasis Cutánea

combinando información metabolómica y SNPs

Juan Pablo Mejía Patiño (Autor Trabajo de Grado)

Nota de Aceptación

Certificamos que el presente Trabajo de Grado Satisface, en alcances y calidad, todos los requisitos que demanda un Trabajo de Grado de Maestría.

DIEGO LUIS LINARES Ph. D

Director

MARIA CONSTANZA PABON BURBANO Ph. D
Jurado

MARIO JULIAN MORA CARDONA
Jurado

Aprobado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana Cali, para optar el título de Magister en Ciencia de Datos.

HERNÁN CAMILO ROCHA NIÑO Ph. D.
Decano Facultad de Ingeniería y Ciencias

JUAN CARLOS MARTÍNEZ ARIAS
Director Posgrados de Ingeniería y Ciencias

Santiago de Cali, 23/02/2024

Acta de Correcciones al Documento de Trabajo de Grado

Santiago de Cali, Febrero 23 de 2024

Autor: Juan Pablo Mejía Patiño

Título del Trabajo de Grado: Predicción del desenlace terapéutico para Leishmaniasis Cutánea combinando información metabolómica y SNIPS

Director: Ph.D. Diego Luis Linares

Como indica el artículo 2.13 de las Directrices para Trabajo de Grado de Maestría, he verificado que el estudiante indicado arriba ha implementado todas las correcciones que los Jurados del Proyecto de Trabajo de Grado definieron que se efectuaran, como consta en el Acta de Evaluación correspondiente.

Firma del Director del Trabajo de Grado

Santiago de Cali, 9 de diciembre del 2023

Doctora
Gloría Inés Álvarez V.
Directora Maestría en Ciencia de Datos
Facultad de Ingeniería y Ciencias
Pontificia Universidad Javeriana de Cali

Asunto: Presentación para evaluación del proyecto aplicado

Cordial Saludo,
Con el fin de cumplir con los requisitos exigidos por la Universidad para optar por el título de Magíster en Ciencia de Datos, nos permitimos presentar a su consideración el proyecto denominado "Predicción del desenlace terapéutico para Leishmaniasis Cutánea combinando información metabolómica y SNPs", el cual fue realizado por el (los) estudiante (s) Juan Pablo Mejía Patiño con código (s) 0220577 pertenecientes a la Maestría en Ciencia de Datos, bajo la dirección de PhD Diego Luis Linares.
El suscrito director del Proyecto Aplicado autoriza para que se proceda a hacer la evaluación de este proyecto, toda vez que ha revisado cuidadosamente el documento y avala que ya se encuentra listo para ser presentado y sustentado oficialmente.

Atentamente,

_____Firma Estudiante(s)_____ _____Firma Director_____
Juan Pablo Mejía Patiño PhD. Diego Luis Linares
C.C. 1.144.088.976 de Cali C.C. _____de _____

Predicción del desenlace terapéutico para Leishmaniasis Cutánea combinando información metabolómica y de SNPs

Juan Pablo Mejía Patiño

Proyecto Aplicado para optar al título de
Magister en Ciencia de Datos

Director
Dr. Diego Linares
Co directora
Dra. María Adelaida Gómez

FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN CIENCIA DE DATOS
SANTIAGO DE CALI, DICIEMBRE 9 DE 2023

FICHA RESUMEN

TÍTULO:

Predicción del desenlace terapéutico para Leishmaniasis Cutánea combinando información metabolómica y SNPs

1. ÁREA DE TRABAJO: Bioinformática
2. TIPO DE PROYECTO (Aplicado, Innovación, Investigación): Investigación
3. ESTUDIANTE(S): Juan Pablo Mejía Patiño
4. CORREO ELECTRÓNICO: juanpmejia@javerianacali.edu.co
5. DIRECCIÓN Y TELEFONO: Calle 16ª # 122B – 150 Apto 402 A
6. DIRECTOR: PhD. Diego Luis Linares
7. VINCULACIÓN DEL DIRECTOR: Profesor de planta Pontificia Universidad Javeriana Cali
8. CORREO ELECTRÓNICO DEL DIRECTOR: dlinares@javerianacali.edu.co
9. CODIRECTORA: PhD. María Adelaida Gómez
10. GRUPO O EMPRESA QUE LO AVALA: DESTINO / CIDEIM
11. OTROS GRUPOS O EMPRESAS: N/A
12. PALABRAS CLAVE: Aprendizaje Supervisado, Leishmaniasis Cutánea, Aprendizaje Automático, Metabolitos, Mutación Genética
13. FECHA DE INICIO: enero 2023
14. DURACIÓN ESTIMADA (En meses): 11 meses
15. RESUMEN: La Leishmaniasis cutánea es una enfermedad presente en múltiples regiones tropicales del mundo, afectando a diversos grupos poblaciones y territorios. América Latina es uno de estos territorios, con la presencia de 15 de sus variedades. Esta enfermedad parasitaria afecta a grupos poblacionales vulnerables que requieren de un tratamiento especializado. Sin embargo, este tratamiento no siempre es exitoso y sus efectos colaterales son, en algunos casos, severos. Teniendo en cuenta esto, es importante contar con herramientas que permitan determinar con un grado alto de confianza el desenlace terapéutico de estos pacientes. Con este objetivo, el presente proyecto busca brindar una predicción sobre el desenlace del tratamiento para la Leishmaniasis Cutánea con un alto grado de confianza, utilizando dos fuentes de datos. Una de información metabolómica y otra de mutaciones genéticas conocidas como "SINGLE NUCLEOTIDE POLYMORPHISMS (SNPs)", junto con técnicas de aprendizaje automático clásicas. Con base en proyectos del grupo DESTINO como antecedentes, se realizaron 18 experimentos aplicando 3 técnicas de

aprendizaje supervisado. De estos, 9 experimentos resultaron en 9 clasificadores base, 6 con cada uno de los conjuntos de datos ya mencionados, y adicionalmente 3 con un nuevo conjunto de datos, originado de la intersección de muestras entre las dos fuentes de datos. Posteriormente, se evalúa su desempeño con métricas como “Accuracy”, “Precision”, “Recall” y “F1 Score”. A partir de esto, se realiza un afinamiento de hiperparámetros de estos clasificadores, usando una técnica de grilla y de nuevo se analizan los resultados con las métricas antes mencionadas. También se experimenta con una técnica de ensamble en cascada, como segunda mecánica para realizar la predicción del tratamiento contra la leishmaniasis. Esto se realiza utilizando los 2 mejores clasificadores que resultan de la fase de afinamiento de modelos. Al finalizar, se obtuvo que un clasificador que mezcla como entradas 7 SNPs, por parte del conjunto de datos de mutaciones genéticas, y 3 metabolitos del conjunto de datos de información metabolómica, obtiene un desempeño superior a los clasificadores con conjuntos de datos separados. Así mismo, el método de ensamble resultó en clasificaciones con un alto nivel de confiabilidad. Esto evidencia, que, combinando fuentes de información diferente bajo dos mecánicas distintas, es posible obtener una herramienta clínica para predecir el desenlace del tratamiento contra la leishmaniasis cutánea.

TABLA DE CONTENIDO

INTRODUCCIÓN 7
1. DEFINICIÓN DEL PROBLEMA 8
2. OBJETIVOS DEL PROYECTO 10
3. MARCO TEÓRICO Y ANTECEDENTES 11
4. PREPARACIÓN Y DISEÑO DEL PROCESAMIENTO DE INFORMACIÓN METABOLÓMICA Y DE SNPs 20
5. ENTRENAMIENTO Y RESULTADOS DE MODELOS 23
6. ANÁLISIS DE RESULTADOS DE MODELOS INICIALES 33
7. INTERFAZ PARA DESPLIEGUE Y CONSUMO DE MODELO ENTRENADO 35
8. CONCLUSIONES 37
9. TRABAJOS FUTUROS 38
10. REFERENCIAS BIBLIOGRÁFICAS 39
A. ANEXO A 41
B. ANEXO B 47
C. ANEXO C 51

LISTA DE FIGURAS

Figura 1: Estructura de 1-octadecanoyl-sn-glycero-3-phosphocholine. 12
Figura 2: Nucleótidos.....13
Figura 3: Single nucleotide polymorphism.....14
Figura 4: (a) Perceptrón simple con dos entradas y salidas. (b) Red Neuronal con una capa oculta, dos entradas y dos salidas.....15
Figura 5: Vectores de Soporte en un espacio bidimensional. 16
Figura 6: Búsqueda de grilla sobre dos parámetros.....17
Figura 7: Arquitectura predictor con fuentes de datos individuales.....23
Figura 8: Arquitectura Predictor con datos mezclados. 24
Figura 9: Arquitectura Ensamble en Cascada.....31
Figura 10: Contrato API para sistema Predictor 35
Figura 11: Resumen Estadístico Conjunto de Metabolómica.....47
Figura 12: Resumen Visual Estadístico Metabolito #1 47
Figura 13: Resumen Visual Estadístico Metabolito #2 48
Figura 14: Resumen Visual Estadístico Metabolito #3 49
Figura 15: Moda para el conjunto de SNPs seleccionados.....49
Figura 16: Interacción con el API + respuesta del servidor (Herramienta: Postman) 51
Figura 17: Implementación de Servidor para API de Predicción con mejor modelo de Gradient Boosting.....52

INTRODUCCIÓN

La Leishmaniasis cutánea es una enfermedad presente en regiones tropicales que afecta a múltiples territorios y grupos poblacionales en múltiples territorios. En este sentido, el Centro Internacional de Entrenamiento e Investigaciones Médicas (CIDEIM) ha enfocado sus esfuerzos en el estudio de enfermedades infecciosas, y en particular con un enfoque en la Leishmaniasis que a su vez impulsó nuevos avances en otras enfermedades. En este marco, y de la mano del Grupo de Investigación DESTINO de la Pontificia Universidad Javeriana, se han venido desarrollando diferentes proyectos que apoyan las dinámicas de investigación sobre esta enfermedad infecciosa, y especialmente, en su tratamiento. Como se presentará en este documento, el tratamiento de esta enfermedad presenta unos desafíos y consideraciones que se pretenden entender y resolver usando la aplicación de ciencia de datos y técnicas de aprendizaje automático.

Como se mencionó antes, el enfoque del trabajo y específicamente de este proyecto en el marco de la colaboración entre el grupo DESTINO y el CIDEIM, fue el de apoyar el proceso de tratamiento de la enfermedad infecciosa conocida como la Leishmaniasis aplicando técnicas de aprendizaje automático, produciendo modelos que fueran capaces de predecir el desenlace de la terapia con un porcentaje de precisión superior a aproximaciones anteriores en estos proyectos. En el presente trabajo se exploraron técnicas clásicas de aprendizaje automático supervisado en conjunto con dos conjuntos de datos proporcionados por el CIDEIM. El primer conjunto de datos relacionado con información metabolómica y el segundo conjunto relacionado con información de SNIPS. Estos marcadores y datos fueron utilizados de manera conjunto para determinar el resultado del desenlace terapéutico de los pacientes que son tratados en esta enfermedad.

El propósito de este proyecto fue se centra en el objetivo del CIDEIM de reducir el impacto de esta enfermedad infecciosa, de manera que sea posible predecir con un grado de confianza el resultado de la terapia previo a aplicarse al paciente. La razón detrás de esto es que el tratamiento en cuestión es un tratamiento con un impacto considerable que afecta la calidad de vida de estas personas. Como se verá en el presente trabajo, muchas veces estos tratamientos no son exitosos y solo es posible saberlo una vez que el paciente fue sometido este tratamiento que puede ser bastante doloroso para el individuo. Con los resultados aquí consignados, se espera brindar una alternativa para el CIDEIM con el fin de mejorar la calidad de vida de las personas afectadas por

esta enfermedad a través del uso de capacidades tecnológicas como el aprendizaje automático y la ciencia de datos. Esto se podrá reflejar en un impacto a la sociedad y al entorno del país.

1. DEFINICIÓN DEL PROBLEMA

PLANTEAMIENTO DEL PROBLEMA

La Leishmaniasis es una enfermedad parasitaria causada por un organismo protozoario y que se manifiesta con diferentes síntomas como: lesiones en la piel, inflamación de mucosas degenerativa e infección visceral. En otras palabras, es una enfermedad que destruye diferentes tejidos en el cuerpo del huésped. Existen 22 variedades de protozoarios capaces de producir la infección, según la OMS. Esta misma entidad señala que 15 de esas variedades se encuentran en las américas. El parásito se transmite mediante la picadura de insectos flebótomos hembras de la familia Lutzomyia, con picaduras durante la noche, cuando estos están activos y lo transmiten al ser humano. [1]

Entre los 10 países que reportan más casos a nivel mundial para la Leishmaniasis cutánea, 4 de ellos se encuentran en el continente americano: Brasil, Colombia, Nicaragua y Perú. La leishmaniasis cutánea es la forma más frecuente de la infección y se manifiesta con lesiones ulcerosas que pueden dejar cicatrices para toda la vida. Actualmente, no existe una vacuna que prevenga la infección de este parásito. Por lo tanto, una vez se produce la infección, esta solo puede ser tratada. [1]

El tratamiento de la Leishmaniasis es un tratamiento bastante severo para el paciente y su efectividad no tiene una alta tasa de efectividad. En [2]se expone que el tratamiento con una sustancia como el antimonio puede ser un tratamiento probablemente exitoso, pero la cantidad de aplicaciones necesarias alcanzan la misma morbilidad que en el escenario en donde la enfermedad no se trata. Adicionalmente, el antimonio es costoso, exceptuando su forma genérica [2]. Dependiendo de la variedad del parásito que causa la infección en el norte de Suramérica, por ejemplo, la leishmaniasis brazilensis o leishmaniasis panamensis, la tasa de curación para tres meses puede oscilar entre el 60 y 90% [2]. Estas sustancias pueden tener grandes impactos en órganos como el corazón, así como otros órganos del paciente. Es en este punto donde el CIDEIM (Centro de Internacional de Entrenamiento e Investigaciones Médicas), se involucra como un actor

en la investigación de la Leishmaniasis para identificar alternativas terapéuticas que sean más efectivas, seguras y fáciles de administrar que las que existen actualmente. La trayectoria del CIDEIM lo han hecho reconocer como un Centro Colaborador de la OMS desde 1992 y un Centro de Referencia para identificar especies de Leishmania. [3]

En el marco de la investigación del CIDEIM, y en conjunto con el grupo DESTINO de la Pontificia Universidad Javeriana, se busca lograr la predicción del desenlace terapéutico y qué características influencian una respuesta positiva frente al tratamiento, aplicando técnicas de Aprendizaje Automático y de analítica de datos. El proceso de decisiones del tratamiento se podría ver beneficiado, mejorando la calidad de vida de los pacientes y evitando incertidumbres en su proceso de recuperación

FORMULACIÓN DEL PROBLEMA

1.2.1 Formulación

¿Cómo realizar la predicción del resultado terapéutico de la Leishmaniasis Cutánea usando fuentes de información metabólica y SNPs de los pacientes del CIDEIM?

1.2.2 Sistematización

¿Cómo se puede mezclar los distintos tipos de información de las bases de datos de información metabólica y las bases de datos de SNPs para poder procesarlas con algoritmos de aprendizaje automático?

¿Cómo usar las fuentes de información para realizar la predicción del desenlace terapéutico en un paciente con técnicas de aprendizaje automático clásicas?

¿De qué manera se puede evaluar la predicción realizada por el modelo desarrollado en el proyecto?

2. OBJETIVOS DEL PROYECTO

2.1 OBJETIVO GENERAL

- Predecir el resultado del tratamiento terapéutico para la Leishmaniasis cutánea, combinando fuentes de información metabólica y SNIPS de diferentes bases de datos del CIDEIM.

2.2 OBJETIVOS ESPECÍFICOS

- Mezclar diferentes tipos de información de múltiples bases de datos como información de SNIPS y metabolómica de manera que esta pueda ser procesada por algoritmos de aprendizaje automático.
- Predecir, usando fuentes de datos de diferentes tipos de información que incluyen información de SNIPS y metabolómica, el desenlace de la terapia de un paciente con Leishmaniasis Cutánea por medio de técnicas clásicas de aprendizaje automático.
- Evaluar la precisión de la predicción del desenlace terapéutico usando información genética y metabólica.

3. MARCO TEÓRICO Y ANTECEDENTES

En el marco del desarrollo del proyecto, se revisaron algunos conceptos biológicos involucrados en el estudio del desenlace terapéutico de los pacientes que han sido tratados por padecer la leishmaniasis cutánea. Por otro lado, se hizo revisión de algunas de las técnicas de aprendizaje automático supervisado clásicas que han sido relevantes en algunos de los procesos de predicción del desenlace terapéutico de esta enfermedad. A continuación, se exponen algunas definiciones alrededor de los diferentes marcadores biológicos como los metabolitos y los. Así mismo, se exploran conceptos el perceptrón multicapa, el concepto de "gradient boosting" y la técnica de máquina de vectores de soporte, como parte de algunas técnicas de aprendizaje supervisado clásicas. Por último, se hizo una revisión de antecedentes en el área de la terapia para la Leishmaniasis cutánea, y sobre algunos resultados que se han tenido en esta área.

3.1. MARCO TEÓRICO

Metabolitos

Una aproximación que se ha tenido a la hora de estudiar el desenlace terapéutico de la enfermedad ya mencionada antes es el uso de la información metabolómica de los pacientes. La información metabolómica contiene datos respecto a los biomarcadores conocidos como metabolitos. Los metabolitos son los productos y los resultados intermedios producidos por la actividad metabólica celular. Esto significa que gran parte de las actividades del cuerpo humano generan como resultado estos metabolitos. Los metabolitos tienen gran cantidad de funciones incluyendo conversión de energía, señales, influencia epigenética [4]. Un uso clínico conocido de estos biomarcadores se da en la medición de la glucosa en el fluido cerebro espinal. En el caso particular de la Leishmaniasis, estos biomarcadores se han utilizado para predicción del éxito de una variedad específica de leishmaniasis. A continuación, se puede observar la estructura bidimensional de uno de estos metabolitos, en específico, del metabolito 1-octadecanoyl-sn-glycero-3-phosphocholine

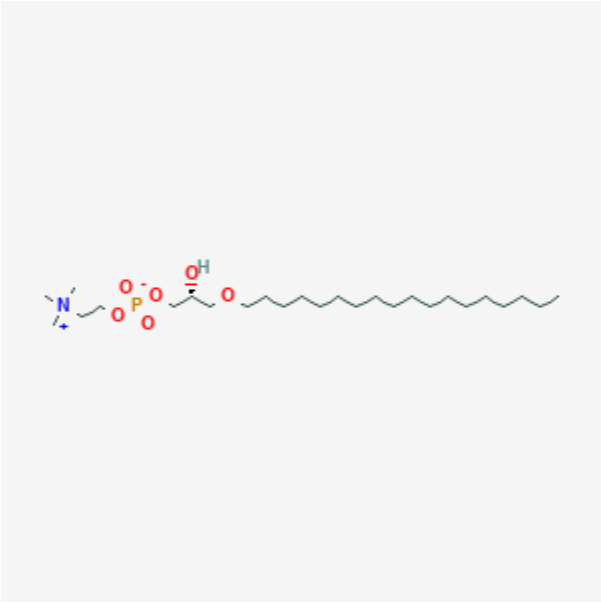

Figura 1: Estructura de 1-octadecanoyl-sn-glycero-3-phosphocholine. Fuente: [5]

El metabolito en la figura 1 es el precursor del mediador inflamatorio de las células cutáneas conocido como "paf (paf-acether)" [6]. Este metabolito está compuesto por carbono, oxígeno, hidrógeno, nitrógeno y fósforo. Su fórmula molecular es $C_{26}H_{56}NO_6P$. Este es uno de muchos metabolitos presentes en los estudios del CIDEIM. Como es posible apreciar, y se reitera, estas son moléculas que son producidas por los distintos procesos del metabolismo humano. Estos productos pueden ser medidos, y en el caso particular del tratamiento de enfermedades, es posible observar el comportamiento de estos metabolitos previo a un tratamiento y posterior a un tratamiento.

Single nucleotide polymorphism (SNPs)

En el cuerpo humano, se encuentra la molécula de Ácido Desoxirribonucleico (ADN), encargada de contener la información genética para el desarrollo y funcionamiento de un organismo [7]. Esta cadena se compone de nucleótidos. Estos son los bloques fundamentales de los ácidos nucleicos. Estos se componen de una molécula de azúcar adherida a un grupo fosfato y una base que contiene nitrógeno [8]. En la figura 2 de [8] se aprecian los nucleótidos como las unidades fundamentales de una cadena de ADN

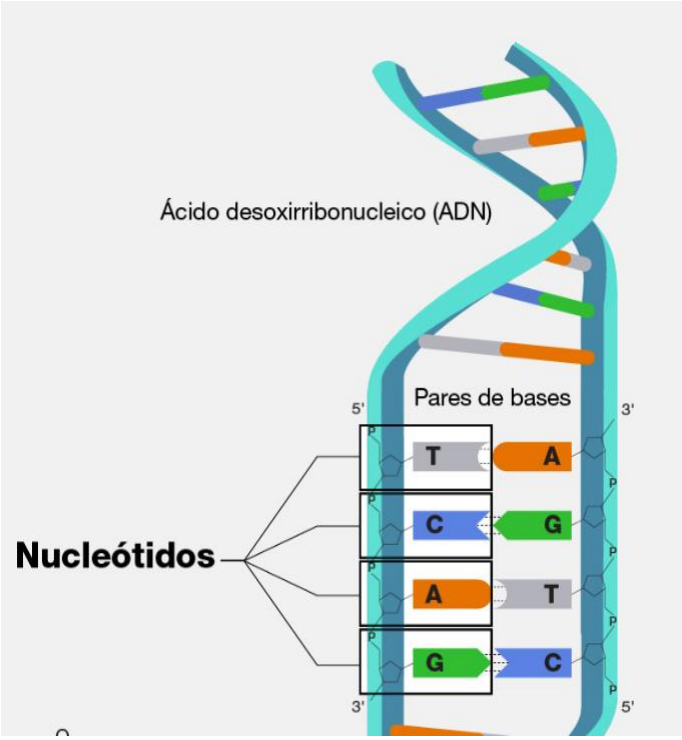

Figura 2: Nucleótidos. Fuente: [8]

Sin embargo, estos nucleótidos en algunos casos sufren de variaciones de un genoma a otro. Esto permite introducir el segundo biomarcador de interés sobre el proyecto, llamado "Single nucleotide polymorphism" o SNPS (pronunciados como SNIPS). Los SNPs se definen como la variante genómica en una posición base del ADN, en otras palabras, son diferencias en la cadena de ADN que se pueden presentar de un individuo a otro. Esto puede verse incluso a nivel de ciertas poblaciones y no solo de un individuo. Los SNPs son, en algunos casos, los responsables de influenciar la salud, las enfermedades, las respuestas a fármacos, entre otros aspectos [9]. En la figura 3 obtenida de [9], se puede apreciar como una de estas diferencias se manifiesta en la cadena de ADN de diferentes individuos

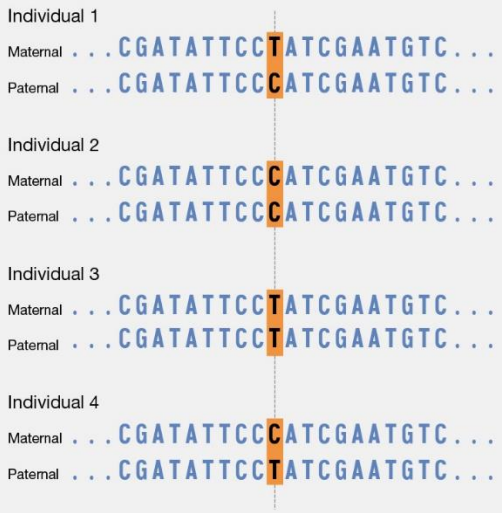

Figura 3: Single nucleotide polymorphism. Fuente: [9]

Aprendizaje supervisado

En el campo de la inteligencia artificial, se tiene el proceso de aprendizaje automático. Para el aprendizaje supervisado, como se describe en [10], se define como el proceso de aprendizaje en donde la información que entrena un modelo está etiquetada, es decir, que esta información contiene ejemplos, así como su correspondiente valor objetivo. Este tipo de aprendizaje automático se utiliza en problemas de predicción o clasificación. Existen varios modelos fundamentales alrededor del aprendizaje supervisado, que sirven para clasificar o predecir. A continuación, se nombran tres de las técnicas utilizadas en aprendizaje supervisado

Perceptrón Multicapa: El perceptrón multicapa es una implementación de lo que es llamado una red neuronal artificial. La idea detrás de esta técnica es tener una red de nodos, llamadas **unidades**, que conectadas permiten propagar un impulso, usando una función de activación y representando las conexiones entre estas unidades con un peso determinado que indica la fuerza de la conexión entre las mismas [11]. En la figura 4, tomada de la literatura de Russell [11] se puede apreciar una estructura de un perceptrón simple y una de una red neuronal con capas ocultas

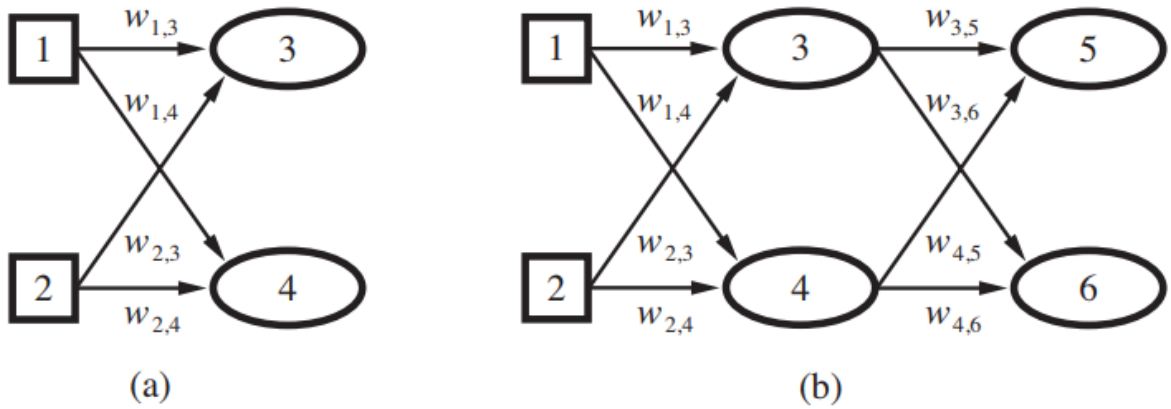

Figura 4: (a) Perceptrón simple con dos entradas y salidas. (b) Red Neuronal con una capa oculta, dos entradas y dos salidas

Fuente: Russell [11]

Es importante señalar que, así como otros modelos, las redes neuronales artificiales con capas ocultas pueden ser sujeto de sobreajuste cuando existen muchos parámetros en el modelo. En otras palabras, pueden no generalizar bien los datos como modelos que tienen pocos parámetros. En el caso de las redes neuronales totalmente conectadas existen dos parámetros que quedan por escoger para determinar la estructura del modelo: la cantidad de capas ocultas y el tamaño de cada capa. En [9] Russell y Norvig indican que el método principal para lograrlo es probar varias combinaciones y elegir el mejor, combinado con técnicas de validación cruzada.

Gradient Boosting: El Gradient boosting es una técnica de ensamble para aprendizaje supervisado que caracteriza por entrenar un modelo sencillo, se evalúa computando el error residual de este predictor, y se entrena un nuevo modelo que intenta predecir el error residual del anterior modelo [12]. Al combinar estos modelos, se genera un modelo más fuerte, y se reduce el error residual del primer modelo. Este proceso se puede repetir de manera iterativa para corregir el error de modelos anteriores y generar nuevos modelos que mejoren de manera incremental el “accuracy” del predictor. En esta técnica, se construye un árbol de decisión para predecir el error residual y hacer el ajuste de cada modelo. Los hiperparámetros claves de este tipo de clasificador normalmente se encuentran en la cantidad de estimadores y la tasa de aprendizaje.

Máquina de Vectores de Soporte: La máquina de vectores de soporte es una técnica supervisada que se considera como una de las técnicas supervisadas más populares [11]. Es especialmente útil cuando no se tiene un conocimiento previo especializado del dominio sobre el que se encuentra un problema. Existen tres propiedades [11] que las hacen una técnica popular:

1. Esta técnica construye un margen de separación máxima, que le permite determinar un límite donde la distancia de separación se maximiza a los ejemplos. Esto se traduce en que pueden generalizarse muy bien.
2. Esta técnica puede separar linealmente con un hiperplano los ejemplos. Sin embargo, es posible embeber los datos en espacios de mayor dimensionalidad usando el truco de kernel para separar datos que no son linealmente separables en el espacio de entrada original.
3. Por último, las máquinas de vectores de soporte, depende de algunas observaciones claves en el conjunto de datos, que se convierten en los vectores de soporte. Esto significa que la máquina no requiere de todas las observaciones para ajustar un separador. Esto hace a la técnica resistente al sobreajuste y permite representar funciones complejas.

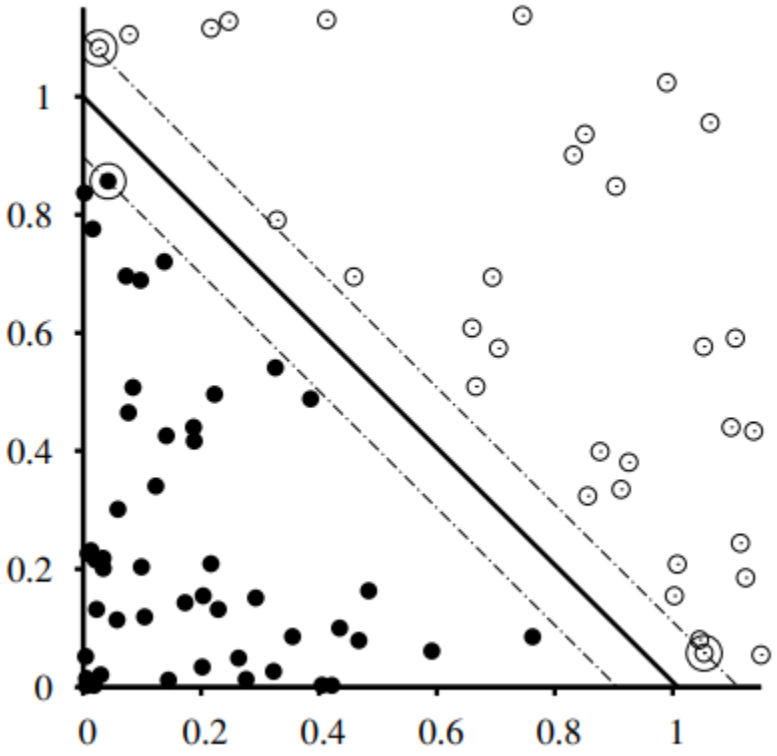

Figura 5: Vectores de Soporte en un espacio bidimensional. Fuente: Russell [11]

En la figura 5, se puede observar los elementos claves para una máquina de vectores de soporte. Las clases para separar son representadas los puntos oscuros y claros. El separador es la línea gruesa, que se encuentra entre las líneas punteadas. Los círculos más grandes interceptados con

las líneas punteadas representan las observaciones más cercanas al separador que sirven como vectores de soporte.

Búsqueda de Grilla

Es una técnica de búsqueda exhaustiva que entrena un modelo con un conjunto de hiperparámetros dado, que permite encontrar la combinación óptima de estos hiperparámetros en el entrenamiento sobre el conjunto de datos [13]. Es una técnica que computacionalmente es costosa si un conjunto de datos es de tamaño considerable. En [13], se establece que, para conjuntos de datos pequeños, es una técnica apropiada y confiable para encontrar el mejor conjunto de hiperparámetros para un modelo. A continuación, se observa una figura de [13], que muestra la aproximación exhaustiva que tiene la técnica de búsqueda de grilla para encontrar la mejor combinación de hiperparámetros.

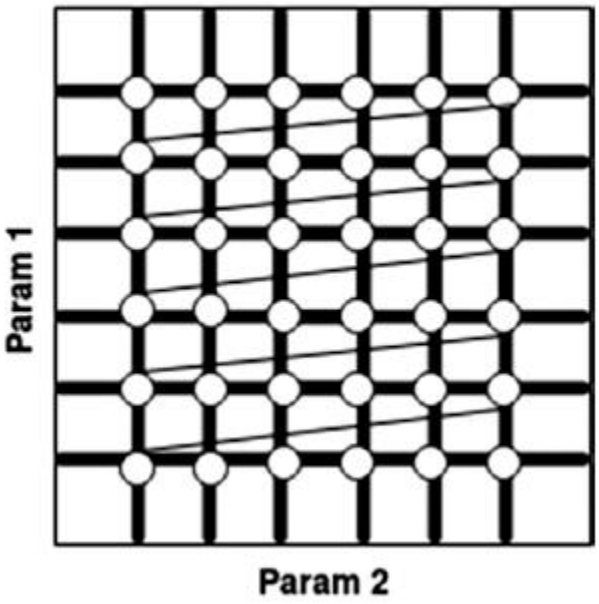

Figura 6: Búsqueda de grilla sobre dos parámetros.

Fuente: Agrawal [13]

Bibliotecas y Herramientas

Las técnicas antes mencionadas presentan diferentes implementaciones en el módulo de scikit-learn para Python. La biblioteca scikit-learn es un módulo de Python, creado en el año 2007, por David Cournapeau, que integra una gran variedad de algoritmos de aprendizaje automático de estado del arte, para aprendizaje supervisado de mediana escala y problemas no supervisados. Su énfasis, como lo indica en [14], es en la facilidad de uso, desempeño, documentación y consistencia en su API, con dependencias mínimas. Algoritmos para Gradient Boosting, Máquina de Vectores de Soporte y Perceptrón Multicapa presentan una serie de implementaciones estables que permiten construir sistemas predictores con el mismo.

3.2. ANTECEDENTES

Este proyecto se planteó con base en la existencia de proyectos en el área, usando técnicas distintas u otras fuentes de información para la predicción. Uno de estos proyectos se presenta en [15], en el año 2021, en Irán, los autores e investigadores buscaban detectar casos que no respondieran positivamente a los tratamientos con drogas estándar en pacientes que sufren la Leishmaniasis Cutánea causada por la sepa *Leishmania Tropica*. La resistencia a los tratamientos es bastante común según se expone en [15], por lo tanto, era clave identificar rápidamente un paciente cuyo tratamiento no estaba respondiendo. Para ello, se involucraron técnicas de aprendizaje automático supervisado con el fin de clasificar a los pacientes entre casos que respondían y casos que no respondían al tratamiento. Se exploraron modelos como K-vecinos más cercanos, máquina de vectores de soporte, perceptrón multicapa, entre otros. Finalmente, el modelo seleccionado fue el clasificador Perceptrón Multicapa, este modelo obtuvo resultados de 87.8%, 90.3% y 0.88% en medidas cómo *Sensitivity, Specifity* y área bajo la curva ROC, respectivamente. Se identificó que la característica que más influenció la respuesta al tratamiento fue la cantidad de tiempo de la lesión, y la que menos afectó fue el género del paciente.

Por otro lado, existen trabajos enmarcados en la alianza con el CIDEIM con el grupo de investigación DESTINO de la Pontificia Universidad Javeriana Cali. En [16] y [17], se desarrollan proyectos con información de SNPs (mutaciones genéticas) y metabolómica respectivamente. En ambos proyectos se realizaba predicción del desenlace terapéutico. En [16], se realizó un proceso de reducción de dimensionalidad y aplicación de técnicas de aprendizaje automático supervisado con resultados de alto desempeño, al aplicarlo en los datos de mutación genética. De manera similar, se tiene que en [17] se aplican técnicas supervisadas clásicas de aprendizaje automático, en donde se resalta el clasificador de vectores de soporte. Con este, en las observaciones

pretratamiento se obtuvo resultado de “accuracy” de 0.77 como el mejor resultado en términos de desempeño para un subconjunto de datos (RFE) pretratamiento. Así mismo, se destaca que la técnica con mejor desempeño en términos de “accuracy” en el mismo subconjunto de datos fue la de regresión logística con 0.85 de “accuracy”. Cabe resaltar que, en el marco del mismo proyecto, también se obtuvieron resultados con una red neuronal artificial para otro subconjunto de datos (ANOVA) de datos pretratamiento con información metabolómica, con un “accuracy” de 0.62. Es importante señalar, que ambos trabajos se desarrollaron con información de población en territorio colombiano.

4. PREPARACIÓN Y DISEÑO DEL PROCESAMIENTO DE INFORMACIÓN METABOLÓMICA Y DE SNPs

En el marco de desarrollo del proyecto, se tuvo a disposición la información de muestras de pacientes del CIDEIM en forma de archivos XLSX y CSV. Sin embargo, algunos de estos datos se encontraban en una disposición estructural que requería preprocesamiento para realizar la experimentación. En particular, el conjunto de datos de información metabolómica presentaba una disposición en sus filas y columnas, que hizo necesario aplicar algunas transformaciones adicionales sobre los datos que fueron suministrados por el centro de investigación.

Cómo se mencionó antes, para lograr la utilización de estos conjuntos de datos, se requirió de una revisión preliminar de su estructura. Esto se evidenció notablemente en el conjunto de datos de metabolómica. El objetivo de estas transformaciones era el de obtener 3 conjuntos de datos a partir de las dos fuentes originales de información: un conjunto de entrenamiento y evaluación con datos metabolómicos, un conjunto de entrenamiento y evaluación con datos de SNPs y un conjunto de entrenamiento y evaluación de datos mixto, que reunía características metabolómicas y de SNPs de un mismo paciente. En otras palabras, para estos últimos, la intersección de ambos conjuntos de datos originales. A continuación, se describen las características seleccionadas y las estrategias utilizadas el procesamiento inicial de estas fuentes de información.

Selección de Metabolitos y SNIPS para procesamiento

Para el conjunto de datos de SNPs, y fundamentado en los resultados presentados en el trabajo de Gómez en [16], se toman los siguientes SNPs para su procesamiento y pruebas. Estos SNPs presentaron un desempeño alto como se evidencia en [16] y se utilizaron como base para la experimentación en el presente proyecto:

- rs10800745
- rs4859120
- rs10061385
- rs573057
- rs11101913
- rs2379991
- rs8042254

Por otro lado, para el conjunto de datos de Metabolitos, y fundamentado en los resultados presentados en el trabajo de Zuluaga en [17], se tomaron los siguientes metabolitos para su procesamiento y pruebas:

- [FA (11:2)] 4_10-undecadiynal

- [PC (18:0)] 1-octadecanoyl-sn-glycero-3-phosphocholine
- D-Erythrose

Transformación

Una vez seleccionados en base al trabajo presentado en [16] y [17], se procedió con varias transformaciones a cada conjunto de datos. Estas transformaciones se mencionan y describen a continuación.

Transformaciones para el conjunto de metabolómica

1. Transposición de Filas y Columnas: Las observaciones de este dataset se encontraban en formas de columnas para cada paciente. Por lo tanto, se debía realizar una transposición para obtener las observaciones en la estructura apropiada.
2. Identificación de tipos de datos de columnas: Posterior a la transposición, se debe validar qué datos son categóricos y si se requiere de codificación o de algún tratamiento para un tipo de dato numérico en las columnas resultantes.
3. Selección y filtrado de muestras para pacientes pretratamiento: El conjunto de datos de metabolitos contenía información de los pacientes pretratamiento y postratamiento, por lo que se debía remover los datos postratamiento. La idea era realizar predicción del resultado terapéutico con datos previos a cualquier tratamiento, y es en este paso donde se realiza esta selección de filas.
4. Renombramiento de llaves de paciente: Se renombraron las llaves del paciente para remover prefijos innecesarios y que no permitían unir la columna del resultado terapéutico que se encontraba en una estructura tabular separada.
5. Mezcla de muestras de pacientes con tabla de ID de muestras y resultado del tratamiento: Se realiza una operación de mezcla en Python, con los ID de muestras para juntar las columnas de metabolitos con el resultado de desenlace terapéutico.
6. Selección de columnas: En este paso se seleccionan las columnas que corresponden a los tres metabolitos de interés para los experimentos.

Transformaciones para el conjunto de SNPs

1. Selección de Columnas: En este paso se seleccionan las columnas que corresponden a los SNPs de interés para los experimentos.
2. Identificación de tipos de datos de columnas: Se validaron qué datos son categóricos y si se requiere de codificación o de algún tratamiento para un tipo de dato numérico en las columnas resultantes.
3. Codificación de columnas categóricas: Se realizó una codificación de las columnas de SNPs con un codificador de etiquetas, debido a que estas columnas son de naturaleza categórica y se hace necesario para alimentar los modelos de los experimentos posteriores.

Para el conjunto de datos de SNPs, la cantidad de transformaciones requeridas fue sustancialmente menor que la del conjunto de metabolitos.

Transformaciones para el conjunto de datos mixto (SNPs + Metabolitos)

Una vez los conjuntos de datos anteriores fueron preparados para el entrenamiento de modelos en una estructura apropiada, se procedió a realizar la construcción del tercer conjunto de datos necesario para la totalidad de los experimentos.

Para este nuevo conjunto de datos, se resolvió usar la mezcla o búsqueda de intersección usando la llave de “Sample ID”, proporcionada por el centro de investigación. Esto permitió generar una intersección que ofrecía 22 muestras para trabajar.

Una vez identificada la intersección, se aplicó la técnica de sobremuestreo sobre el conjunto de datos resultante para realizar un balance de clases. De esta manera, al realizar la división para el entrenamiento, se tendría un conjunto más equilibrado y óptimo para entrenar y validar.

División de los datos test-train

Para dividir los datos en conjuntos de entrenamiento y pruebas, se planteó una división de 70% para datos de entrenamiento y el 30% restante para realizar pruebas. La partición de los datos se realizó para cada conjunto de datos desde el principio y se mantuvo la misma para cada uno de los experimentos que se realizaron y que se presentarán en este mismo documento en capítulos posteriores.

5. ENTRENAMIENTO Y RESULTADOS DE MODELOS

Para lograr la predicción utilizando técnicas clásicas de aprendizaje supervisado, se plantearon doce experimentos con los conjuntos de datos antes mencionados. Se planteó un experimento por cada técnica con cada conjunto de datos. Se escogieron tres técnicas de aprendizaje supervisado, cuya implementación está disponible dentro del paquete de herramientas de scikit-learn. Estas técnicas son, en su respectivo orden: Perceptrón Multicapa, "Gradient Boosting", Máquina de Vectores de Soporte.

A continuación, se puede ver la figura 7, donde se presenta el pipeline de predicción al entrenar los modelos con las dos fuentes de datos individuales, realizando predicción del desenlace terapéutico:

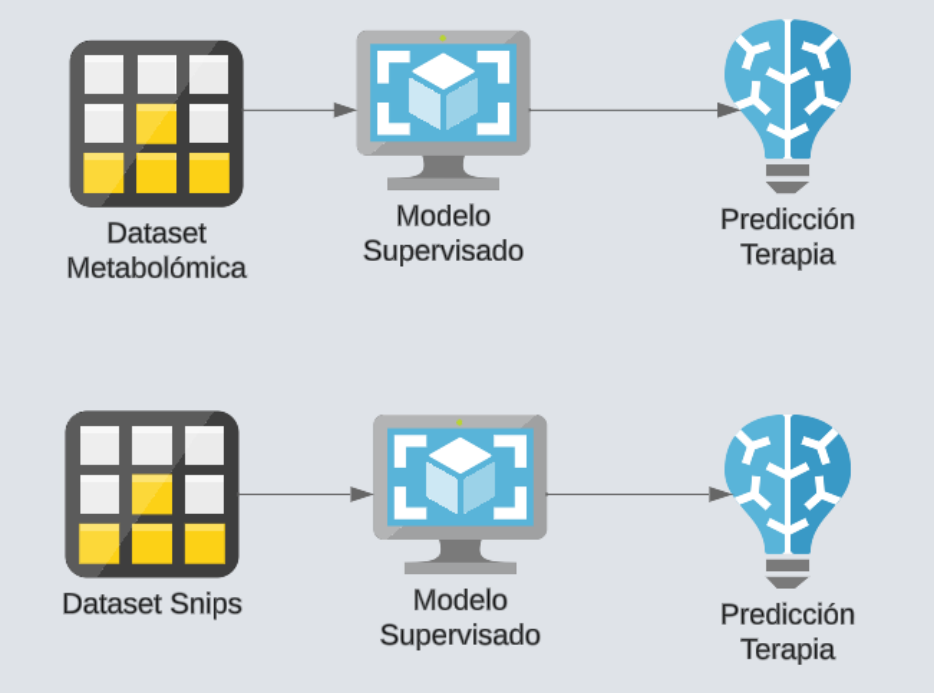

Figura 7: Arquitectura predictor con fuentes de datos individuales. Fuente: Autor

Por otra parte, se puede apreciar en la figura 8 cómo la intersección de los datos sirve como una tercera fuente de datos que también alimenta un modelo supervisado como parte de los doce experimentos planteados:

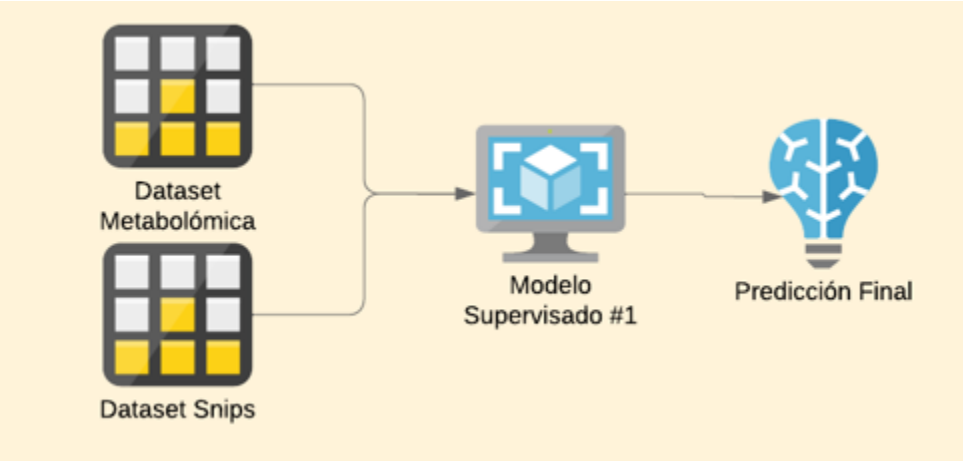

Figura 8: Arquitectura Predictor con datos mezclados. Fuente: Autor

Modelos Base

Para iniciar el entrenamiento de clasificadores, se partió una serie de modelos base con el fin de evaluar su desempeño inicial, para así proceder con el afinamiento de parámetros. A continuación, se presentan los resultados iniciales de los nueve experimentos con modelos base.

TABLA 1: Resultados Peceptrón Base

Dataset	Accuracy Score	Precision Score	Recall Score	F1 Score
Metabolitos	0.500	0.250	0.500	0.333
SNPs	0.600	0.619	0.600	0.583
Metabolitos + SNPs	0.500	0.25	0.500	0.333

TABLA 2: Resultados Gradient Boosting Base

Dataset	Accuracy Score	Precision Score	Recall Score	F1 Score
Metabolitos	0.863	0.873	0.863	0.866
SNPs	0.772	0.786	0.772	0.777
Metabolitos + SNPs	0.863	0.885	0.863	0.846

TABLA 3: Resultados Máquina de Vectores de Soporte Base

Dataset	Accuracy Score	Precision Score	Recall Score	F1 Score
Metabolitos	0.500	0.250	0.500	0.333
SNPs	1	1	1	1
Metabolitos + SNPs	0.875	0.900	0.875	0.873

Afinamiento de Hiperparámetros con el método de búsqueda de grilla

Una vez se obtuvieron los modelos base, se procedió a realizar entrenamiento de clasificadores, haciendo ajuste de sus hiperparámetros para mejorar el desempeño. Para tal fin, se decidió aplicar el método de búsqueda de grilla, también conocido como "Grid Search" en las tres técnicas para obtener los mejores clasificadores. La búsqueda de grilla se centró en encontrar diferentes configuraciones de hiperparámetros que generaran el mejor clasificador. Se escogió esta técnica, a pesar de su costo computacional, porque los conjuntos de datos y la cantidad de columnas no requería una capacidad de cómputo alta, así como el tiempo requerido para aplicar la búsqueda.

Para cada una de las técnicas, se aplicó el método de búsqueda de grilla tomando la partición de entrenamiento de cada conjunto de datos. Como se describe en el capítulo 4, el criterio para definir el conjunto de entrenamiento fue tomar el 70% del conjunto de datos de metabolitos, el 70% del conjunto de datos de SNPs y el 70% del conjunto de datos mezclados de información metabolómica y SNPs. En adelante, estas particiones serán señaladas añadiendo la palabra "train" al final de cada conjunto de datos. Durante la aplicación de la búsqueda de grilla, se alimentó el método con la partición de "train" de cada fuente de datos. A estos conjuntos se les aplicó una validación cruzada, usando el parámetro k con valor de 7. De esta manera el método seleccionó la mejor combinación combinación de parámetros, luego de evaluar distintas combinaciones de estos hiperparámetros. Para la validación final de los estimadores obtenidos en la búsqueda, se tenía la partición del 30% restante de cada conjunto de datos, que será referida de aquí en adelante con el nombre del conjunto de datos anexando la palabra "test". La intención de esta segunda partición era tener muestras que nunca habían sido observadas por los mejores clasificadores. Esto permitía evaluar de mejor manera el desempeño de estos. De esta manera, se podía comparar los clasificadores en términos de las mismas particiones de los conjuntos de datos.

Para la red neuronal, se hizo el afinamiento con "Grid Search" en dos pasos. El primer paso permitió ajustar la arquitectura de las capas ocultas del perceptrón. Esto orientado a determinar la mejor estructura para el perceptrón. En el segundo paso, se afinaron otros parámetros del perceptrón como el "solver", la función de activación y la tasa de aprendizaje. Con este método, se buscó reducir el tiempo de entrenamiento y de la búsqueda de hiperparámetros. Sin embargo, se debe resaltar nuevamente que la cantidad de registros y columnas involucradas no presentaban un tiempo de cómputo considerable. Pero teniendo en cuenta que estos conjuntos de datos

pueden crecer en un trabajo futuro, se propuso realizarlo de esta manera, contemplando algún desafío en el tiempo de cómputo necesario.

Por otro lado, para la máquina de vectores de soporte y la técnica de "gradient boosting", también se aplicó una búsqueda de grilla, con el propósito de realizar el afinamiento y obtención del mejor modelo aplicando estas técnicas. Diferente al perceptrón, aquí no se dividió en dos pasos la búsqueda de hiperparámetros. Se hizo una búsqueda por modelo.

Hiperparámetros Perceptrón Multicapa

A continuación, se presentan los resultados de la búsqueda de hiperparámetros para el perceptrón con cada uno de los conjuntos de datos de entrenamiento que se utilizaron como base para los experimentos.

TABLA 4: Hiperparámetros Perceptrón Multicapa para Metabolitos (train)

Hiperparámetro	Valores Búsqueda	Valor Óptimo
Hidden Layer Sizes	• (10,) • (10,5) • (10,5,3)	• (10,)
Activation	• 'relu' • 'logistic' • 'tanh'	• 'logistic'
Solver	• 'sgd' • 'adam'	• 'sgd'
Learning Rate	• 0.01	• 0.01

TABLA 5: Hiperparámetros Perceptrón Multicapa para SNPs (train)

Hiperparámetro	Valores Búsqueda	Valor Óptimo
Hidden Layer Sizes	• (10,) • (10,5) • (10,5,3)	• (10,)
Activation	• 'relu' • 'logistic' • 'tanh'	• 'logistic'

Solver	• 'sgd' • 'adam'	• 'sgd'
Learning Rate	• 0.01	• 0.01

TABLA 6: Hiperparámetros Perceptrón Multicapa para Metabolitos y SNPs (train)

Hiperparámetro	Valores Búsqueda	Valor Óptimo
Hidden Layer Sizes	• (10,) • (10,5) • (10,5,3)	• (10,)
Activation	• 'relu' • 'logistic' • 'tanh'	• 'tanh'
Solver	• 'sgd' • 'adam'	• 'adam'
Learning Rate	• 0.01	• 0.01

Hiperparámetros Gradient Boosting

Para la técnica de gradient boosting también se presentan los resultados de la búsqueda de hiperparámetros con cada uno de los conjuntos de datos de entrenamiento que se utilizaron como base para los experimentos:

TABLA 7: Hiperparámetros Gradient Boosting para Metabolitos (train)

Hiperparámetro	Valores Búsqueda	Valor Óptimo
Número de estimadores	• 50 • 100 • 150	• 100
Máxima profundidad	• 1 • 2	• 2
Estado Aleatorio	• 0	• 0

Learning Rate	• 0.01	• 0.01

TABLA 8: Hiperparámetros Gradient Boosting para SNPs (train)

Hiperparámetro	Valores Búsqueda	Valor Óptimo
Número de estimadores	• 50 • 100 • 150	• 50
Máxima profundidad	• 1 • 2	• 2
Estado Aleatorio	• 0	• 0
Learning Rate	• 0.01	• 0.01

TABLA 9: Hiperparámetros Gradient Boosting para Metabolitos y SNPs (train)

Hiperparámetro	Valores Búsqueda	Valor Óptimo
Número de estimadores	• 50 • 100 • 150	• 150
Máxima profundidad	• 1 • 2	• 1
Estado Aleatorio	• 0	• 0
Learning Rate	• 0.01	• 0.01

Hiperparámetros Máquina de Vectores de Soporte

Por último, se presentan los resultados de la técnica de optimización de hiperparámetros para la búsqueda de grilla con cada conjunto de datos de entrenamiento:

TABLA 10: Hiperparámetros Máquina de Vectores de Soporte para Metabolitos (train)

Hiperparámetro	Valores Búsqueda	Valor Óptimo
Factor de regularización	• Rango: Logspace(-5,7,20)	• 29763.5
Kernel	• 'linear' • 'poly' • 'rbf'	• 'poly'

TABLA 11: Hiperparámetros Máquina de Vectores de Soporte para SNPs (train)

Hiperparámetro	Valores Búsqueda	Valor Óptimo
Factor de regularización	• Rango: Logspace(-5,7,20)	• 1.128
Kernel	• 'linear' • 'poly' • 'rbf'	• 'linear'

TABLA 12: Hiperparámetros Máquina de Vectores de Soporte para Metabolitos y SNPs (train)

Hiperparámetro	Valores Búsqueda	Valor Óptimo
Factor de regularización	• Rango: Logspace(-5,7,20)	• 545559.478

Kernel	• 'linear' • 'poly' • 'rbf'	• 'rbf'

Resultados con afinamiento de Hiperparámetros

Una vez se completaron las búsquedas de hiperparámetros, se obtuvieron resultados para nueve experimentos utilizando las particiones de evaluación para cada conjunto de datos. En este momento, se evaluaron los clasificadores afinados y los resultados se presentan en tres tablas donde se discrimina cada conjunto de datos y las respectivas métricas de desempeño.

Para la técnica de la red neuronal artificial, el perceptrón multicapa, se pueden ver las siguientes métricas, discriminadas por su respectivo conjunto de datos

TABLA 13: Resultados Perceptrón Multicapa

Dataset	Accuracy Score	Precision Score	Recall Score	F1 Score
*Metabolitos **(test)***	0.500	0.250	0.500	0.333
*SNPs **(test)***	0.909	0.919	0.909	0.902
*Metabolitos + SNPs **(test)***	0.500	0.25	0.500	0.333

Para la técnica optimizada de gradient boosting, se pueden ver las siguientes métricas, discriminadas por su respectivo conjunto de datos:

TABLA 14: Resultados Gradient Boosting

Dataset	Accuracy Score	Precision Score	Recall Score	F1 Score
*Metabolitos **(test)***	0.600	0.619	0.600	0.583
*SNPs **(test)***	0.727	0.759	0.727	0.737
*Metabolitos + SNPs **(test)***	1.0	1.0	1.0	1.0

Para la técnica de clasificador de vectores de soporte, se pueden ver las siguientes métricas, discriminadas por su respectivo conjunto de datos:

TABLA 15: Resultados Máquina de Vectores de Soporte

Dataset	Accuracy Score	Precision Score	Recall Score	F1 Score
Metabolitos ***(test)***	0.600	0.619	0.600	0.583
SNPs ***(test)***	0.863	0.859	0.863	0.859
Metabolitos + SNPs ***(test)***	0.750	0.833	0.750	0.733

Modelos ensamblados en Cascada

Durante este proyecto se planteó experimentar con otra mecánica, diferente a la de mezclar las dos fuentes de información directamente. Esta fue el ensamble de diferentes modelos. Específicamente, se implementó un sistema predictor en cascada. Esta técnica se implementó utilizando dos de los mejores modelos resultantes posterior al proceso de afinamiento. En este escenario, los dos mejores modelos para los conjuntos de datos individuales fueron: el clasificador de "gradient boosting", entrenado con la información metabolómica y el perceptrón multicapa entrenado con la información de SNPs. En la figura 9 se puede observar la estructura en cascada del ensamble que se implementó.

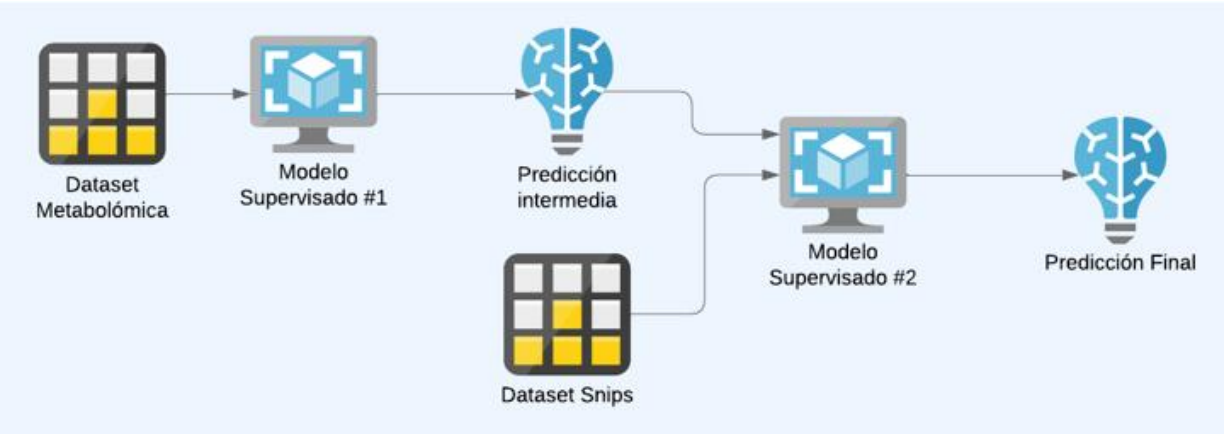

Figura 9: Arquitectura Ensamble en Cascada. Fuente: Autor

A partir de esta arquitectura de ensamble, se realizaron dos experimentos adicionales. En el primer experimento, se estableció un umbral con la probabilidad de clase del modelo más débil.

En este caso, si la probabilidad de clase del clasificador de información metabolómica no superaba 0.95, se realizaba una segunda predicción con el clasificador para la información de mutaciones (SNPs). Esto permitió decidir en el pipeline cuándo aplicar esta cascada. Cuando la probabilidad de la predicción por la clase no supera un umbral establecido, se recurre a un segundo modelo, más fuerte.

Para la técnica de ensamble basado en un umbral, se obtuvo el siguiente resultado:

TABLA 16: Resultados Ensamble con Umbral

Accuracy Score	Precision Score	Recall Score	F1 Score
1.0	1.0	1.0	1.0

Por otra parte, en el segundo experimento se estableció un promedio ponderado para decidir el resultado de la predicción. A diferencia del primer experimento, en este los dos clasificadores realizan su predicción. Sin embargo, a cada clasificador se le dio un peso ponderado en esa predicción. De esta manera, al clasificador que se observó con mejor desempeño, se le dio una mayor ponderación, con un peso ponderado de 0.7. este clasificador es el perceptrón multicapa para el conjunto de datos de SNPs. Por otro lado, se le asignó 0.3 como peso ponderado al clasificador más débil, el clasificador de “gradient boosting” para la información metabolómica.

Para la técnica de ensamble con promedio ponderado, se obtuvo el siguiente resultado:

TABLA 17: Resultados Ensamble con Promedio Ponderado

Accuracy Score	Precision Score	Recall Score	F1 Score
1.0	1.0	1.0	1.0

6. ANÁLISIS DE RESULTADOS DE MODELOS INICIALES

Los resultados presentados en el capítulo anterior sirven como insumo para realizar un proceso de análisis respecto a su desempeño y su utilidad a la hora de realizar predicciones para el desenlace terapéutico. En este capítulo se revisan las técnicas que destacaron por su desempeño en cada uno de los conjuntos de datos. De igual manera, se discuten algunas de las razones para el desempeño presentado en los mejores modelos de cada técnica y se revisa si la mezcla de conjunto de datos resultó beneficiosa.

Para el conjunto de metabolitos, se pudo apreciar que la técnica de máquina de vectores de soporte tuvo un desempeño igual al del perceptrón multicapa. En términos de "accuracy" se observó que el modelo no tuvo mucho éxito en clasificar correctamente si un paciente es curado o no. Este alcanza apenas un 60.0%. A pesar de esto, con respecto al modelo base, que obtuvo 50.0% de "accuracy", este clasificador tuvo mejor desempeño al aplicar la búsqueda de hiperparámetros y afinarlo. Sin embargo, la mejora con respecto al modelo base fue de solo un 10%, lo que indica que la técnica no es muy prometedora en la experimentación realizada.

En segundo lugar, en el conjunto de SNPs, se pudo apreciar que el perceptrón multicapa tuvo un desempeño alto con un porcentaje de "accuracy" del 90.9% con el conjunto de evaluación. Con respecto al clasificador base, el "accuracy" de este clasificador mejoró en un 30.9%. La mejora también se presentó en las métricas de "precisión", "recall" y "F1 score". Esta es una mejora sustancial. Adicionalmente, el porcentaje de "recall" indica que este clasificador es muy eficaz identificando los casos verdaderamente positivos en términos de pacientes que se consideran con desenlace terapéutico positivo, alcanzando un porcentaje de 90.2%. Sin embargo, es importante señalar que la máquina de vectores de soporte, en su modelo base, obtuvo un desempeño del 100% en su evaluación y se resalta como un resultado llamativo.

Finalmente, para el conjunto de la intersección, el resultado con la técnica de gradient boosting fue extraordinario con el conjunto de evaluación. Se obtuvo un "accuracy", "precision" y "recall" del 100%. La técnica de gradient boosting fue la de mejor desempeño entre todas las técnicas luego de realizar el afinamiento con el conjunto de entrenamiento de la intersección. Con respecto al modelo base, su desempeño se incrementó un 13.7% en términos de "accuracy". Así mismo, este superó el desempeño del perceptrón multicapa que fue el mejor modelo con únicamente la información del conjunto de entrenamiento de SNPs por aproximadamente un 9.1% en términos de "accuracy". Esto permite afirmar que la mezcla de información metabolómica y de SNPs puede resultar beneficiosa para incrementar el desempeño al predecir el desenlace terapéutico. Esto benefició también a una técnica en específico por encima de las otras. También es importante señalar que la técnica de gradient boosting tuvo un desempeño en "accuracy" superior en un 25% y 50% sobre la máquina de vectores de soporte y perceptrón multicapa respectivamente. Esto también resulta interesante, a pesar de que ele perceptrón multicapa tuvo un excelente desempeño con el conjunto compuesto únicamente por SNPs, su desempeño se degradó sustancialmente con la mezcla de información.

Por otro lado, al revisar los resultados del pipeline de predicción en cascada, se puede afirmar que los resultados obtenidos son de alto desempeño. Esto refuerza la idea de que una mezcla de información, incluso bajo otra mecánica a través de la predicción en cascada, resulta beneficiosa para la predicción del desenlace terapéutico de la leishmaniasis cutánea usando información metabolómica y SNPs. Con el modelo ensamblado en cascada se obtuvo puntajes en las cuatro métricas de 1.0. Este desempeño es igual que el alcanzado al mezclar las fuentes de datos directamente hallando la intersección de los datos, luego de hacer el proceso de preprocesamiento.

Teniendo en cuenta lo anterior, se puede afirmar que la mezcla de las fuentes de datos con un único modelo resulta ser la mejor opción. A pesar de que tiene el mismo desempeño que el pipeline ensamblado en cascada, esta opción requiere de entrenar con una sola técnica un conjunto de datos. Esto puede reflejarse en menos tiempo en términos de implementación, sin la necesidad de entrenar diferentes clasificadores con distintas técnicas. Así mismo, no es necesario definir un ponderación o umbral para tomar una acción al momento de determinar el resultado de una predicción en el desenlace terapéutico.

7. INTERFAZ PARA DESPLIEGUE Y CONSUMO DE MODELO ENTRENADO

Como parte de los objetivos específicos del presente proyecto, se buscaba implementar una interfaz que permitiera utilizar los modelos entrenados como una API, con el fin de ampliar el horizonte de uso más allá de un Jupyter Notebook. Esto presentó la posibilidad de profundizar en un método para la puesta en producción de un clasificador ya entrenado, que pudiera incorporarse a otras aplicaciones o a interfaces gráficas que el personal del CIDEIM puede usar.

Fue con este objetivo en mente que se implementó un API (Application Programming Interface), utilizando el framework "Flask". De esta forma, fue posible desplegar un servidor web simple de manera local, que alojara el predictor en forma de un API. Este servidor solo tenía un "endpoint" que ofrecía un método POST para realizar la predicción, teniendo como entrada una estructura JSON que representaba la observación de un paciente. Esta entrada es una entrada simplificada que aísla al usuario de conocer los detalles de implementación en la forma de estructurar los datos de entrada que utiliza el modelo. A continuación, se presenta un contrato que permite ver la estructura JSON que se envía en el cuerpo de un "request" al API que se desarrolló en el marco del proyecto.

```
{
      'met_1': float,
      'met_2': float,
      'met_3': float,
      'rs10800745': int,
      'rs4859120': int,
      'rs10061385':int,
      'rs573057': int,
      'rs11101913': int,
      'rs2379991': int,
      'rs8042254': int

}
```

Figura 10: Contrato API para sistema Predictor

Para los parámetros met_1, met_2 y met_3, se tienen que corresponden respectivamente a las mediciones de los metabolitos del paciente en su respectivo orden: [FA (11:2)] 4_10-undecadiynal, [PC (18:0)] 1-octadecanoyl-sn-glycero-3-phosphocholine, D-Erythrose. En el caso de los parámetros que le siguen, se corresponden con los SNIPS de la misma denominación. Estas son columnas que corresponden a datos categóricos que se deben codificar como un número entero.

Para ofrecer el clasificador que ha sido entrenado en un API, se realizó un proceso de escritura de los modelos en memoria al disco como un archivo ". pkl", utilizando la biblioteca "pickle" en

Python. De esta forma se consiguió servir un API de manera local con un modelo previamente entrenado en el marco del proyecto, que extiende su uso por fuera de un “Jupyter Notebook”. [18]

8. CONCLUSIONES

Como conclusión principal, es posible afirmar que se pudo entrenar y evaluar de manera exitosa modelos que tuvieron un desempeño alto al momento de combinar dos fuentes de datos distintas, de información metabolómica y de SNIPS. El proyectó demostró que es posible combinar estas fuentes de datos que son diferentes y obtener desempeños altos y, en algunos casos, superiores a los clasificadores usados con una sola fuente de datos. Así mismo, a pesar de que la cantidad de muestras no era una cantidad alta, la proporción y los datos encontrados en la intersección del conjunto hicieron viable la aplicación de técnicas de aprendizaje supervisado para predecir el desenlace terapéutico. Adicionalmente, se puede concluir que la técnica de gradient boosting es una técnica competitiva, ya que se obtuvo un puntaje de "accuracy", "precision", "recall" y "F1 score" de 1.0 con el conjunto de evaluación. No obstante, la técnica de máquina de vectores de soporte tuvo un buen desempeño que plantea la posibilidad de realizar mayor exploración con esta técnica para mejorar su desempeño.

En la misma línea, se pudo desarrollar un sistema predictor ensamblado en cascada, que es útil para predecir el desenlace terapéutico, valiéndose de los mejores clasificadores hallados durante los experimentos. Estos clasificadores, gradient boosting para metabolitos y perceptrón multicapa para SNPs, presentaron un alto desempeño en las cuatro métricas evaluadas al integrarse en cascada. Ambos experimentos obtuvieron un puntaje de 1.0 para "accuracy", "precision", "recall" y "F1 score". Esto demuestra otro camino viable para integrar ambas fuentes de datos en un sistema de predicción con un alto grado de confianza. Teniendo información metabolómica y de SNPs, con predictores independientes, es posible predecir el resultado de la terapia. Sin embargo, se puede concluir que la mezcla de las fuentes de datos con un único modelo resulta ser la opción más viable debido a que requiere de entrenar con una sola técnica un conjunto de datos ya mezclado. Esto es necesario considerarlo en un ejercicio de ingeniería y de proyectos porque los recursos en un proyecto de ciencia de datos son limitados. Tanto el tiempo, como el poder de cómputo, así como sus costos, llevan a pensar que entrenar más técnicas para construir una cascada podría resultar en mayores costos, sin que el desempeño sea significativamente mayor al que se ha alcanzado al mezclar primero los datos y entrenar un modelo como gradient boosting.

Por último, se pudo implementar un API simple en un servidor local, que permite el consumo de uno de los modelos que se alimenta de la mezcla de ambas fuentes de información, permitiendo que el horizonte de uso de esta herramienta vaya más allá de un cuaderno de "Jupyter" y pueda integrarse en un sistema de mayor complejidad o con una interfaz de usuario para los investigadores del CIDEIM. Esto facilita el ingreso de parámetros y la obtención de la predicción, sin añadir complejidad con la ejecución de un cuaderno de "Jupyter". De esta manera, se están ofreciendo más herramientas al CIDEIM para expandir su trabajo en el mejoramiento de la calidad de vida de las personas que sufren de la Leishmaniasis Cutánea, y que se enfrentan a un tratamiento donde el resultado muchas veces no es positivo, y no es fácil predecirlo.

9. TRABAJOS FUTUROS

Como trabajo futuro, y teniendo en cuenta los resultados alcanzados durante este trabajo, se recomiendan dos rutas que tienen valor en el marco de continuar esta experimentación. En primer lugar, se plantea extender la experimentación con un conjunto de datos con más observaciones. Esto con el fin de conseguir mejores estimadores y evitar tener clasificadores que puedan sufrir de exceso de confianza. Esto permitirá reducir la incidencia del tamaño del conjunto de datos en los resultados de los experimentos. Aunque se realizó experimentación utilizando la técnica de sobremuestreo para contrarrestar los efectos del tamaño de los conjuntos de datos, es posible explorar con un conjunto de datos más grande, y de ser necesario, aplicando técnicas de sobremuestreo que generen valor a la hora de entrenar los clasificadores.

Por otra parte, se recomienda extender y escalar la funcionalidad del API implementado. El servidor implementado actualmente solo permite realizar la predicción de una sola observación. Como trabajo futuro, se recomienda explorar la implementación de un API que permita procesar varias observaciones y retornar sus predicciones como una colección. Esto mejoraría la usabilidad del endpoint y podría escalarse para su uso en un aplicativo mucho más complejo.

10. REFERENCIAS BIBLIOGRÁFICAS

[1] OMS, «Leishmaniasis - OPS/OMS | Organización Panamericana de la Salud,» [En línea]. Available: https://www.paho.org/es/temas/leishmaniasis. [Último acceso: 8 Diciembre 2022].

[2] H. W. Murray, J. D. Berman, C. R. Davies y N. G. Saravia, «Advances in Leishmaniasis,» 2005.

[3] CIDEIM, «Leishmaniasis,» CIDEIM, [En línea]. Available: https://cideim.org.co/cideim/en/our-research/researchareas/leishmaniasis.html. [Último acceso: 8 Diciembre 2022].

[4] C. Teunissen, C. Verheul y E. Willemse, Handbook of Clinical Neurology, Elsevier, 2018.

[5] National Center for Biotechnology Information, «PubChem Compound Summary for CID 2733532, 1-O-Octadecyl-SN-glycero-3-phosphocholine,» [En línea]. Available: https://pubchem.ncbi.nlm.nih.gov/compound/2733532#section=2D-Structure. [Último acceso: 06 12 2023].

[6] Y. Denizot, L. Michel, Y. Thomas, M. Heslan y J. Benveniste, «Production of paf-acether by human epidermal cells,» [En línea]. Available: https://pubmed.ncbi.nlm.nih.gov/2230220/. [Último acceso: 26 Diciembre 2023].

[7] National Human Genome Research Institute, «DEOXYRIBONUCLEIC ACID (DNA),» 2023. [En línea]. Available: https://www.genome.gov/genetics-glossary/Deoxyribonucleic-Acid. [Último acceso: 5 Diciembre 2023].

[8] National Human Genome Research Institute, «NUCLEOTIDE,» [En línea]. Available: https://www.genome.gov/genetics-glossary/Nucleotide. [Último acceso: 5 Diciembre 2023].

[9] National Human Genome Research Institute, «SINGLE NUCLEOTIDE POLYMORPHISMS (SNPS),» National Human Genome Research Institute, 4 Diciembre 2023. [En línea]. Available: https://www.genome.gov/genetics-glossary/Single-Nucleotide-Polymorphisms. [Último acceso: 5 Diciembre 2023].

[10] J. Li, «Regression and Classification in Supervised Learning,» de *Proceedings of the 2nd International Conference on Computing and Big Data - ICCBD 2019*, 2019.

[11] S. Russell y P. Norvig, Artificial Intelligence: A Modern Approach, Pearson, 2009.

[12] A. Ihler, «Ensembles: Gradient Boosting (CS273a: Lecture),» University of California Irvine, 2012.

[13] T. Agrawal, Hyperparameter Optimization in Machine Learning, Bangalore: Apress, 2021.

[14] F. Pedregosa y et al., «Scikit-learn: Machine Learning in Python,» *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[15] M. Bamorovat, I. Sharifi, E. Rashedi, A. Shafian, F. Sharifi, A. Khosravi y A. Tahmouresi, «A novel diagnostic and prognostic approach using artificial neural networks,» *PLOS ONE,* vol. 16, 5 Mayo 2021.

[16] C. Gomez, «Clasificación de pacientes con leishmaniasis basado en mutaciones genéticas por polimorfismo de nucleótido único (SNP) usando técnicas de Machine Learning,»

Pontificia Universidad Javeriana Cali, Cali, 2022.

[17] S. Zuluaga, «Aprendizaje de Máquina Aplicado a la Predicción del Éxito del Tratamiento de la Leishmaniasis,» Pontificia Universidad Javeriana, Cali, 2022.

[18] J. Mejia, «Graduation Project for Magister Degree in Data Science (PUJ),» 2024. [En línea]. Available: https://github.com/juanpmejia/lh-predict-mixdata. [Último acceso: 13 Enero 2024].

A. ANEXO A

Tabla 18: Conjunto Train Metabolitos Post-Procesamiento

	[FA (11:2)] 4_10-UNDECADIYNAL	[PC (18:0)] 1-octadecanoyl-sn-glycero-3-phosphocholine	D-Erythrose	Respuesta
7	6151,598	3498168	21076,74	0
18	6859,235	2478269	19918,31	0
24	8140,19	4084666	25134,54	1
8	7477,694	2941742	18779,54	0
15	12314,85	2659492	24669,38	0
4	6467,406	2608760	21806,93	0
29	9101,834	3890887	20600,38	1
30	10393,34	3181904	10908,08	1
32	12100,13	4166258	20114,57	1
5	6090,529	2728504	18697,39	0
21	8730,236	2939934	22974,8	0
1	7498,572	2095392	29008,21	0
25	7043,212	2925104	13485,19	1
3	5711,174	3480736	27113,84	0
17	9344,885	3143997	16958,97	0
6	8140,348	2704562	10457,12	0
23	11795,28	3826207	11902,33	1
11	6717,1	2540003	21041,63	0

0	5127,423	3331903	26200,63	0
14	10261,58	2473965	21860,58	0
13	5453,397	2736192	22796,08	0
19	9076,019	1901419	21559,58	0
12	7430,245	2576174	24022,45	0

Tabla 19: Conjunto Test Metabolitos Post-Procesamiento

	[FA (11:2)] 4_10-UNDECADIYNAL	**[PC (18:0)] 1-octadecanoyl-sn-glycero-3-phosphocholine**	**D-Erythrose**	**Respuesta**
9	2851,816	2783345	35565,45	0
16	10416,82	2540316	28369,78	0
26	8299,794	2926051	19561,63	1
2	5152,68	3397793	20116,25	0
10	6891,813	3462315	11893,82	0
22	7865,299	1815553	9871,237	1
28	6869,92	2458349	17242,36	1
31	9098,916	3279609	12785,97	1
20	8700,357	3211458	21648,4	0
27	6311,634	3348933	14710,53	1

Tabla 20: Conjunto Train SNPs Post-Procesamiento

	rs10800745	**rs4859120**	**rs10061385**	**rs573057**	**rs11101913**	**rs2379991**	**rs8042254**	**Respuesta**
11	0	1	1	1	2	2	0	0

61	1	0	1	1	0	1	1	1
16	0	1	2	1	2	2	0	0
24	0	0	0	0	0	0	2	1
12	0	1	1	0	0	2	1	0
15	0	1	1	0	1	1	1	0
41	0	1	1	1	1	2	0	0
71	1	0	0	1	0	2	1	1
28	1	0	0	1	2	0	2	1
50	1	0	0	2	1	0	2	0
2	1	0	1	2	0	2	0	0
60	2	0	1	1	1	1	1	1
7	1	0	0	0	0	0	2	1
10	0	1	1	0	2	1	1	0
67	1	0	0	2	0	1	1	1
27	1	1	0	2	1	1	1	1
31	1	0	0	2	0	0	2	1
9	0	2	1	1	0	2	1	0
33	0	0	0	2	2	1	1	1
53	0	1	0	2	0	0	2	1
69	0	0	0	2	1	0	2	1
65	0	0	1	0	0	1	1	0
63	0	1	2	0	0	1	1	1
32	1	0	0	2	1	1	2	1
43	0	0	2	1	2	0	2	0
14	0	1	1	1	1	1	1	0
30	0	0	0	2	1	1	0	1
36	1	0	1	2	0	0	2	1
66	1	0	1	2	0	1	1	0
56	0	0	1	2	0	1	1	1
26	1	0	1	1	1	0	2	1
19	0	0	2	2	0	1	1	1
62	0	1	0	2	1	0	2	1
5	1	0	1	1	0	0	2	1
42	0	1	1	0	1	1	1	0
46	0	0	0	1	0	2	0	0
4	1	0	0	1	0	1	1	1
70	1	0	0	1	0	1	2	1
59	1	0	0	1	0	2	0	1
49	1	1	0	0	1	1	2	0
48	2	0	0	2	0	1	1	1
68	0	0	0	0	2	2	0	1

38	1	0	0	1	0	0	2	1
58	2	0	0	1	0	1	1	1
25	0	0	1	1	1	1	2	1
20	0	0	0	1	0	1	1	1
40	0	1	2	0	1	2	0	0
39	1	0	0	2	0	1	1	1
45	0	1	2	1	2	1	1	0
44	0	1	0	1	2	1	1	0

Tabla 21: Conjunto Test SNPs Post-Procesamiento

	RS10800745	**rs4859120**	**rs10061385**	**rs573057**	**rs11101913**	**rs2379991**	**rs8042254**	**Respuesta**
37	1	0	0	1	0	1	1	1
55	1	0	0	2	1	0	2	1
21	0	0	0	0	0	1	2	1
57	2	2	0	1	0	1	1	1
47	0	1	1	0	1	0	2	0
54	0	0	1	2	0	0	2	1
35	0	0	1	0	0	0	2	1
52	0	0	0	2	1	1	1	1
3	0	1	2	0	2	2	0	0
29	1	0	1	1	0	0	2	1
64	0	1	0	2	2	1	1	0
1	1	0	1	1	1	1	1	0
6	1	0	0	2	1	1	2	1
8	0	1	0	1	0	0	2	1
0	0	1	0	1	2	1	1	0
34	1	0	1	2	1	0	2	1
22	2	0	1	1	0	0	2	1
23	1	1	0	0	1	0	2	1
51	1	1	1	1	0	1	1	1
17	2	1	1	2	1	0	2	1
13	0	1	2	0	1	2	0	0
18	1	0	0	2	0	1	1	1

Tabla 22: Conjunto Train Metabolitos+SNPs Post-Procesamiento

	[FA (11:2)] 4_10-UNDECADIYNAL	[PC (18:0)] 1-octadecanoyl-sn-glycero-3-phosphocholine	D-Erythrose	rs1080075	rs485910	rs1006135	rs573057	rs1110193	rs237991	rs8042254	Respuesta
14	6151,598	3498168	21076,74	0	0	0	1	0	1	1	1
1	7043,212	2925104	13485,19	0	1	1	0	0	2	1	0
21	6859,235	2478269	19918,31	1	0	0	2	1	1	2	1
18	8730,236	2939934	22974,8	0	1	0	1	0	0	2	1
0	6869,92	2458349	17242,36	0	1	1	1	1	2	0	0
24	9076,019	1901419	21559,58	1	0	0	0	0	0	2	1
20	12314,85	2659492	24669,38	1	0	1	1	1	0	2	1
13	12314,85	2659492	24669,38	1	0	1	1	1	0	2	1
3	8299,794	2926051	19561,63	0	1	2	0	1	2	0	0
6	6869,92	2458349	17242,36	0	1	1	1	1	2	0	0
9	7043,212	2925104	13485,19	0	1	1	0	0	2	1	0
23	12314,85	2659492	24669,38	1	0	1	1	1	0	2	1
5	8140,19	4084666	25134,54	0	1	1	1	2	2	0	0
15	8730,236	2939934	22974,8	0	1	0	1	0	0	2	1
12	8140,19	4084666	25134,54	0	1	1	1	2	2	0	0
2	10393,34	3181904	10908,08	0	1	1	0	1	1	1	0
17	10416,82	2540316	28369,78	1	0	0	1	0	1	1	1

8	8299,794	2926051	19561,63	0	1	2	0	1	2	0	0

Tabla 23: Conjunto Test Metabolitos+SNPs Post-Procesamiento

	[FA (11:2)] 4_10-UNDECADIYNAL	[PC (18:0)] 1-octadecanoyl-sn-glycero-3-phosphocholine	D-Erythrose	rs1080074 5	rs4859120	rs1006138 5	rs573057	rs1110191 3	rs2379991	rs8042254	Respuesta
19	7477,694	2941742	18779,54	0	0	0	0	0	1	2	1
4	6869,92	2458349	17242,36	0	1	1	1	1	2	0	0
7	10393,34	3181904	10908,08	0	1	1	0	1	1	1	0
11	10393,34	3181904	10908,08	0	1	1	0	1	1	1	0
16	9076,019	1901419	21559,58	1	0	0	0	0	0	2	1
10	10393,34	3181904	10908,08	0	1	1	0	1	1	1	0
25	9076,019	1901419	21559,58	1	0	0	0	0	0	2	1
22	6090,529	2728504	18697,39	0	0	2	2	0	1	1	1

B. ANEXO B

Conjunto de Metabolitos

```
data_met.describe()
```

Metabolite	[FA (11:2)] 4_10-undecadiynal	[PC (18:0)] 1-octadecanoyl-sn-glycero-3-phosphocholine	D-Erythrose
count	33.000000	3.300000e+01	33.000000
mean	7875.312848	2.973877e+06	20086.445934
std	2140.594294	5.779245e+05	5937.484457
min	2851.815674	1.815553e+06	9871.237305
25%	6467.405762	2.576174e+06	16958.966800
50%	7498.571777	2.926051e+06	20600.380860
75%	9098.916016	3.348932e+06	22974.802730
max	12314.852540	4.166258e+06	35565.453130

Figura 11: Resumen Estadístico Conjunto de Metabolómica. Fuente: Autor

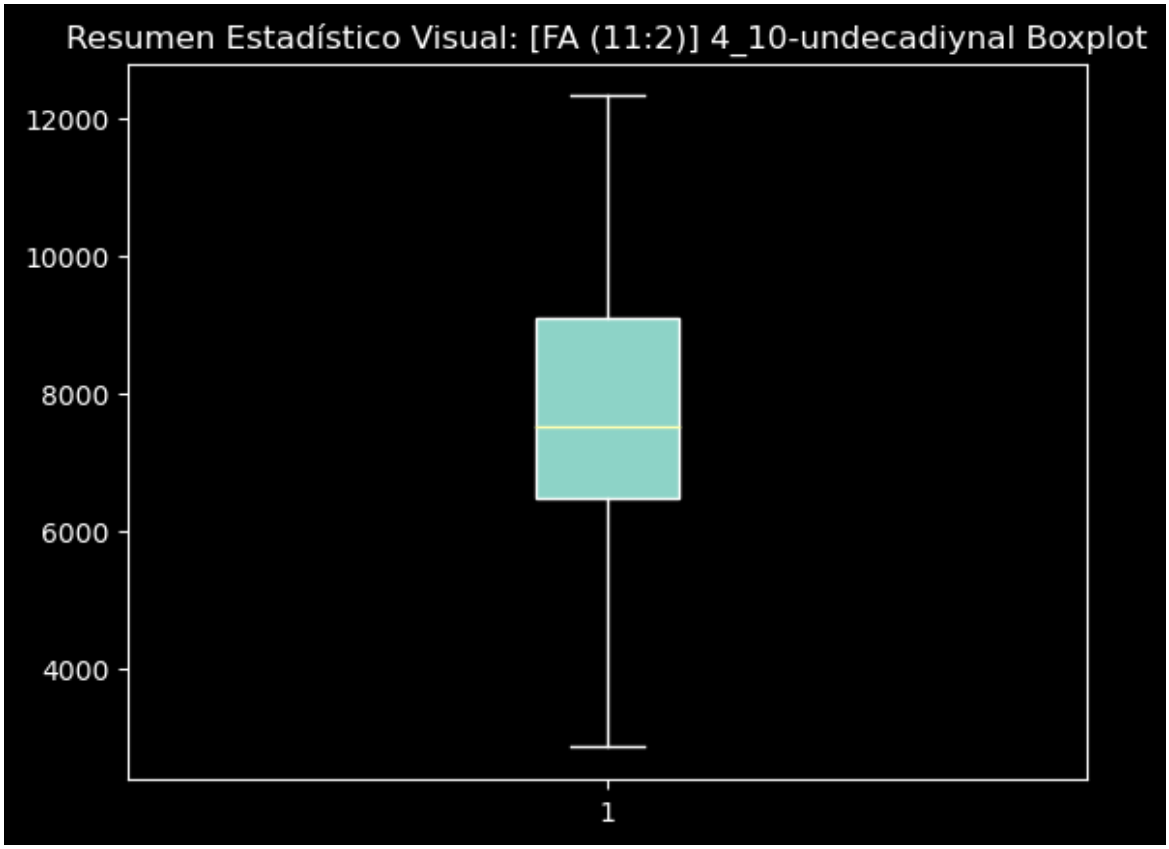

Figura 12: Resumen Visual Estadístico Metabolito #1. Fuente: Autor

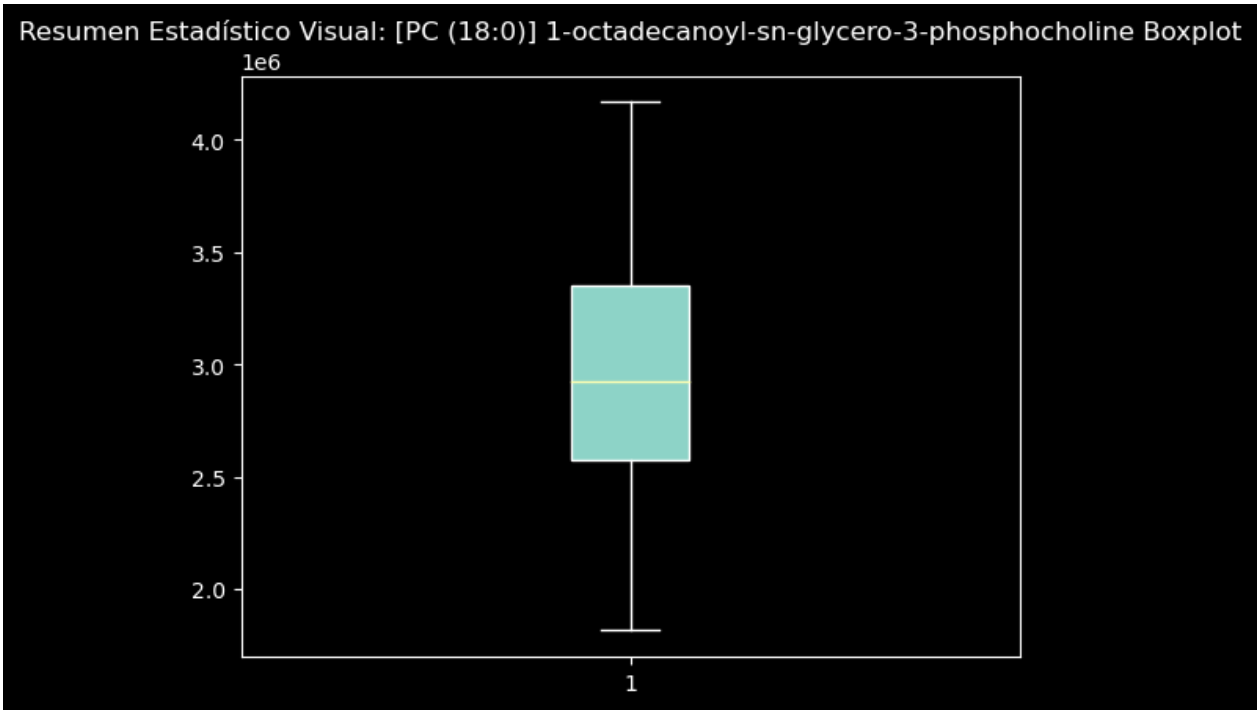

Figura 13: Resumen Visual Estadístico Metabolito #2. Fuente: Autor

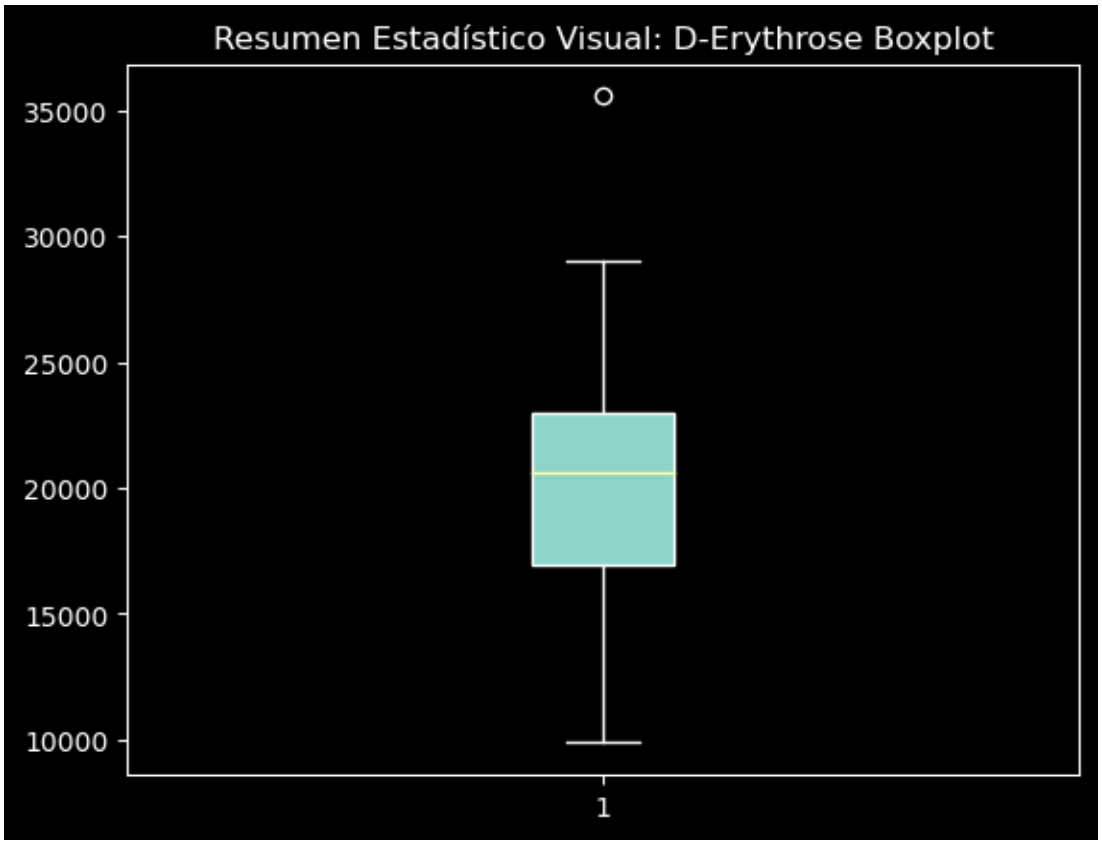

Figura 14: Resumen Visual Estadístico Metabolito #3. Fuente: Autor

Conjunto de SNPs

```
data_snips.mode()
```

	Sample ID	rs10800745	rs4859120	rs10061385	rs573057	rs11101913	rs2379991	rs8042254
0	GT1012	AA	AA	AA	TG	CC	TC	AG

Figura 15: Moda para el conjunto de SNPs seleccionados. Fuente: Autor

C. ANEXO C

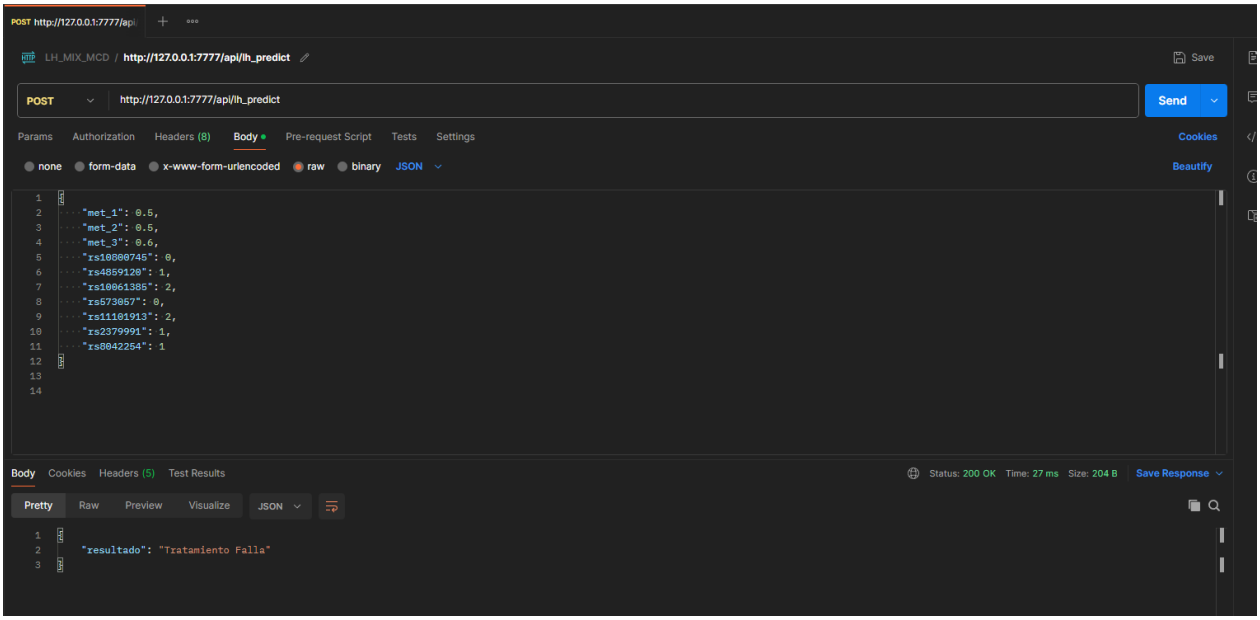

Figura 16: Interacción con el API + respuesta del servidor (Herramienta: Postman). Fuente: Autor

```
#This code has been implemented for project needs based on Script presented
in course
#CS570 from University of Illinois Springfield, by Mike Bernico (2015)
from flask import Flask, request
import numpy as np
import pickle
import pandas as pd
app=Flask(__name__)
# Loading the LH models again, by filename
filename='gradientboosting_clf_2023-12-07T21_mix.pkl'
loaded_clf = pickle.load(open(filename, 'rb'))

@app.route('/api/lh_predict', methods=['POST'])

def make_prediction():
  #Parse the JSON data in the POST Requests and ignore MIME type
  data=request.get_json(force=True)

  #Get the feature values to make the prediction
  req_pred=[data['met_1'],
            data['met_2'],
            data['met_3'],
            data['rs10800745'],
```

```
            data['rs4859120'],
            data['rs10061385'],
            data['rs573057'],
            data['rs11101913'],
            data['rs2379991'],
            data['rs8042254']]
  input_pd=pd.DataFrame(np.array(req_pred).reshape(1,-1),
                        columns=['[FA (11:2)] 4_10-undecadiynal',
                                 '[PC (18:0)] 1-octadecanoyl-sn-glycero-3-
phosphocholine',
                                 'D-Erythrose',
                                 'rs10800745',
                                 'rs4859120',
                                 'rs10061385',
                                 'rs573057',
                                 'rs11101913',
                                 'rs2379991',
                                 'rs8042254'
                                 ])
  pred_result=loaded_clf.predict(input_pd)

  response=[]
  if(pred_result[0]==0):
    response={'resultado':'Tratamiento Falla'}
  else:
    response={'resultado':'Tratamiento Exitoso'}
  return response

if __name__=='__main__':
  app.run(port=7777, debug = True)
```

Figura 17: Implementación de Servidor para API de Predicción con mejor modelo de Gradient Boosting. Fuente: Autor