Pontificia Universidad Javeriana Cali
Facultad de Ingeniería.
Ingeniería de Sistemas y Computación.
Proyecto de Grado.

Predicción del desenlace terapéutico de la leishmaniasis utilizando aprendizaje automático sobre SNPs

Alvaro Jose Bertin Sanchez
Santiago Caicedo Rojas

Directores:
Dra. Gloria Inés Álvarez Vargas
Dr. Diego Luis Linares Ospina

Noviembre 21, 2023

Resumen

Machine learning has enabled significant advancements in the field of medicine, particularly in predicting the outcome of disease treatment. This case study explores machine learning models and techniques to predict the success or failure of leishmaniasis treatment based on Single Nucleotide Polymorphisms (SNPs) in DNA sequences. This is crucial because leishmaniasis treatment can have adverse effects for the human body, making it essential to predict whether individuals with leishmaniasis should undergo treatment.

Unsupervised machine learning techniques were employed for the selection of the most significant SNPs. Subsequently, supervised learning techniques were utilized for prediction, and the performance of the model was assessed. This comprehensive approach aims to determine the efficacy of leishmaniasis treatment and whether individuals should or should not undergo the prescribed regimen.

Índice general

1. Descripción del Problema **9**

1.1. Planteamiento del problema . . . 9

1.1.1. Formulación . . . 10

1.1.2. Subpreguntas . . . 10

1.1.3. Objetivos . . . 10

1.1.4. Justificación . . . 10

1.1.5. Alcance y Limitaciones . . . 10

1.1.6. Entregables . . . 11

2. Desarrollo del Proyecto **13**

2.1. Marco de Referencia . . . 13

2.1.1. Marco teórico . . . 13

2.1.2. SNPs . . . 14

2.1.3. Señalización celular . . . 14

2.1.4. Aprendizaje supervisado y no supervisado . . . 14

2.1.5. Modelos de Machine learning usados . . . 14

2.1.6. Métricas de evaluación . . . 15

2.1.7. Búsqueda de hiperparametros . . . 16

2.1.8. Técnicas de evaluación . . . 18

2.1.9. Silhouette Score . . . 19

2.1.10. Método del codo . . . 19

2.1.11. Estrategia de selección de representantes en Clusterización . . . 19

2.1.12. Estimador de máxima verosimilitud . . . 19

2.1.13. Akaike information criterion (AIC) . . . 20

2.1.14. Bayesian information criterion (BIC) . . . 20

2.2. Trabajos relacionados . . . 20

3. Preprocesamiento de los datos **23**

3.1. Dataset inicial . . . 23

3.2. Transformaciones . . . 23

3.2.1. Codificación . . . 23

3.2.2. Codificación binaria única . . . 24

3.3. Asignación de valores faltantes . . . 24

4. Exploración de subconjuntos de SNPs **25**
4.1. Modelos de clusterización . . . 25
4.1.1. Estrategias de selección de representantes . . . 25
4.2. Preprocesamiento . . . 25
4.3. Clusterización . . . 26
4.3.1. Kmeans . . . 26
4.3.2. GMM . . . 27
4.4. Subconjuntos finales . . . 29

5. Evaluación de los subconjuntos de SNPs **31**
5.1. Técnicas de evaluación . . . 31
5.2. Entrenamiento con subconjunto 0 . . . 32
5.2.1. Iteración 1 . . . 32
5.2.2. Iteración 2 . . . 34
5.3. Evaluación de los subconjuntos . . . 36
5.3.1. Análisis de Resultados de Random Forest . . . 37
5.3.2. Análisis de Resultados de SVM . . . 38
5.3.3. Análisis de Resultados de ANN . . . 39
5.3.4. Conclusión análisis de modelos . . . 41
5.4. Mejora en el entrenamiento con nuevo subconjunto . . . 41
5.4.1. Iteración 1 . . . 41
5.4.2. Iteración 2 . . . 43
5.5. Evaluación de los subconjuntos con los nuevos parámetros . . . 44

6. Análisis de resultados **47**
6.1. Subconjuntos representativos . . . 47
6.2. Análisis y mejores modelos . . . 48

7. Conclusiones **53**
7.1. Trabajo futuro . . . 54

Bibliografía **55**

Capítulo 1

Descripción del Problema

1.1. Planteamiento del problema

La leishmaniasis es una enfermedad que afecta frecuentemente a la población colombiana, se presenta de forma endémica en las zonas rurales produciendo las tres formas clínicas de la enfermedad, las cuales son, la forma cutánea, la forma mucosa y la forma visceral. Esto se debe a que no hay acceso a distintos tratamientos o medicamentos, generando que la enfermedad se propague con facilidad y causando mayor riesgo a la población a medida que pasa el tiempo.

A pesar de que en los últimos años se ha usado el mismo tratamiento para la cura de la leishmaniasis [1], hoy por hoy se desconoce el motivo del por qué algunas personas logran curarse y otras no al usar el tratamiento, logrando curarse solo el 70 % de las personas que padecen cualquier tipo de esta enfermedad, mientras que el otro 30 % no logra curarse.

La leishmaniasis puede causar una destrucción parcial o completa de las membranas de la nariz y la boca que pueden causar discapacidad grave, en caso que sólo presenten leishmaniasis de forma cutánea. Para la forma visceral, causa episodios irregulares de fiebre, perdida de peso, hepatoesplenomegalia, anemia y si no son tratados rápidamente, en el 90 % de los casos, causa la muerte. Por último, para la forma cutánea, produce en su mayoría, lesiones ulcerosas que dejan cicatrices por toda la vida, siendo esta ultima, la forma mas frecuente que se produce en las personas [2].

Entre los problemas que actualmente existen con respecto a la leishmaniasis, es que no se sabe por qué algunos pacientes logran curarse, mientras que otros no lo logran al someterse a un tratamiento médico. Ante esta situación, es necesario encontrar las características de los pacientes que no se logran curar. Esto, con el fin de que otras personas con estas mismas características no se sometan a un tratamiento médico que solo les va a dejar consecuencias negativas en su cuerpo. Para lo anterior, se tiene como objetivo hacer uso de modelos de aprendizaje automático, con los que se hará una aproximación para seleccionar un subconjunto de SNP's de pacientes que no se lograron curar, y así poder predecir el desenlace terapéutico de un paciente según estos SNP's. Un SNP es una variación única del ADN en una persona, en otras palabras, una combinación única de un nucleótido.

Por esta razón, para este trabajo se plantea encontrar la manera de analizar y encontrar similitudes, cadenas en común que tengan las personas que se logran curar y que no se logren curar desde su SNP.

1.1.1. Formulación

¿Como identificar SNPs de interés que contribuyan a la predicción del desenlace terapéutico de la leishmaniasis utilizando modelos de aprendizaje automático?

1.1.2. Subpreguntas

- ¿Cómo preparar el corpus de SNPs para poder procesarlo?
- ¿Cómo seleccionar subconjuntos de SNPs utilizando técnicas de aprendizaje automático?
- ¿Cómo evaluar la calidad de los subconjuntos de SNP escogidos para hacer la predicción del desenlace terapéutico?

1.1.3. Objetivos

Objetivo general: Identificar SNPs de interés que contribuyan a la predicción del desenlace terapéutico de la leishmaniasis utilizando modelos de aprendizaje automático

Objetivos específicos:

- Preparar el corpus de SNPs para poder procesarlo
- Seleccionar subconjuntos de SNPs utilizando técnicas de aprendizaje automático
- Evaluar la calidad de los subconjuntos de SNPs escogidos para hacer la predicción del desenlace terapéutico

1.1.4. Justificación

El lograr predecir el desenlace terapéutico de la leishmaniasis ayudaría a reducir el numero de personas que sufren las consecuencias de un tratamiento ineficaz, y se estaría ahorrando tiempo y dinero.

El proyecto es realizable ya que tiene como soporte al Centro Internacional de Entrenamiento e Investigaciones Médicas (CIDEIM), se cuenta con previas investigaciones sobre subconjuntos de SNPs donde se ha logrado bajar el numero de SNPs de 700.000 a 400, y ahora se realizará buscando como objetivo subconjuntos de máximo 7-8 SNPs, lo cual se puede lograr con los conocimientos que se cuentan y con los recursos disponibles

1.1.5. Alcance y Limitaciones

Para este proyecto se trabajará con una base de datos proporcionada por el CIDEIM con datos de 72 pacientes y 400 SNPs por cada paciente, donde se buscara reducir a subconjuntos de máximo 7-8 SNPs.

1.1.6. Entregables

- Documento de tesis
- Script en python donde se realizó el proceso de selección y evaluación de subconjuntos
- Listado de los subconjuntos de SNPs escogidos

Capítulo 2

Desarrollo del Proyecto

2.1. Marco de Referencia

2.1.1. Marco teórico

Actualmente, en el tratamiento de la leishmaniasis, se usan comúnmente medicamentos como los antimoniales pentavalentes en dos formulaciones dependiendo del tipo de leishmaniasis, estos son antimoniato de N-metil glucamina y estibogluconato de sodio. También, medicamentos como miltefosina, anfotericina B y la anfotericina B liposomal hacen parte de otras opciones terapéuticas, pero, ninguna erradica la infección y las consecuencias adversas que esta enfermedad conlleva [3].

Con respecto a lo anterior, hoy por hoy, se buscan alternativas para combatir la leishmaniasis de tal forma que el efecto en esta sea eliminarla y poder obtener una mejoría en relación a las consecuencias posteriores, de hecho, una alternativa que se está estudiando es la búsqueda de terapias alternativas con el uso de productos naturales, combinando el conocimiento empírico difundido en la población con el uso de plantas medicinales [4].

Otra posible alternativa, es el estudio de asociación del genoma completo (GWAS), este es un enfoque de investigación que se emplea para estudiar las variantes genómicas en busca de similitudes que se presenten con frecuencia en personas con alguna enfermedad o rasgo característico en comparación con aquellos que no tienen la enfermedad o el rasgo [5].

Una investigación hecha con el uso del enfoque GWAS, es titulada "Post-GWAS methodologies for localisation of functional non-coding variants: ANGPTL3"[1] localización de variantes no codificables funcionales identificando rasgos comunes o cadenas comunes en múltiples conjuntos de datos de todo el genoma con datos extraídos de laboratorios en donde se demostró los efectos exitosos y específicos del alelo sobre las interacciones ADN - proteína en línea con los efectos sobre los niveles de LDL -C y la expresión del ARN que se causan por problemas de salud como colesterol alto y triglicéridos.

El grupo Destino junto con el CIDEIM ha trabajado en varios proyectos de investigación con el uso de aprendizaje de máquina aplicado a la predicción del resultado del tratamiento de la leishmaniasis durante varios años.

Con base en lo anterior, este proyecto de investigación hará continuación a las investigaciones o los trabajos anteriores hecho por el grupo Destino junto con el CIDEIM. Específicamente, usando el enfoque de investigación GWAS para identificar variantes comunes de SNP's de varias personas afectadas por la leishmaniasis y se lograron curar en comparación con las variantes comunes de las personas que no lograron curarse.

2.1.2. SNPs

En bioquímica, un SNP (polimorfismo de nucleótido único) es una variación genómica única en el ADN de una persona. Los SNP en un genoma influyen en la salud, enfermedad y respuesta a los fármacos.

2.1.3. Señalización celular

La señalización celular es un proceso crucial que permite a las células comunicarse y coordinar sus funciones en el organismo. Comienza cuando una molécula de señal se une a un receptor en la superficie de la célula, desencadenando una cascada de eventos bioquímicos que transmiten la señal al interior de la célula. Esta señalización regula una amplia variedad de funciones celulares, desde el crecimiento y la diferenciación hasta la respuesta a estímulos ambientales y la supervivencia.

2.1.4. Aprendizaje supervisado y no supervisado

El aprendizaje supervisado es una técnica utilizada en el campo del aprendizaje automático (machine learning). Se basa en el uso de un conjunto de datos de entrenamiento etiquetados, donde cada ejemplo de datos tiene una etiqueta o respuesta conocida. El objetivo del aprendizaje supervisado es aprender una función o modelo que pueda predecir correctamente las etiquetas de nuevos ejemplos no etiquetados.

El aprendizaje no supervisado es otra técnica utilizada en el campo del aprendizaje automático. A diferencia del aprendizaje supervisado, en el aprendizaje no supervisado no se proporcionan etiquetas o respuestas conocidas en los datos de entrenamiento. El objetivo principal del aprendizaje no supervisado es encontrar patrones, estructuras o relaciones intrínsecas en los datos sin la guía de etiquetas predefinidas.

2.1.5. Modelos de Machine learning usados

K-Means y Gaussian Mixture Models (GMM) se escogen para la clusterización debido a su capacidad para identificar patrones en datos no supervisados. A diferencia de otros modelos, K-Means y GMM no hacen suposiciones rígidas sobre la forma de los clusters, siendo efectivos en la captura de estructuras más complejas y distribuciones no lineales.

Random Forest, Support Vector Machines (SVM) y Redes Neuronales Artificiales (ANN) son seleccionados para evaluar clusters, ya que poseen la capacidad de manejar datos complejos y descubrir patrones no lineales en espacios de menor dimensión. Random Forest se destaca por su capacidad para identificar relaciones no lineales y manejar conjuntos de datos con alta dimensionalidad, ofreciendo una perspectiva robusta sobre la estructura de los clusters. SVM, al emplear funciones de kernel, puede mapear datos a espacios de características más complejas, permitiendo la identificación eficiente de fronteras no lineales entre clusters. Por último, las Redes Neuronales Artificiales, al aprender representaciones no lineales a través de capas ocultas, son capaces de modelar relaciones complejas entre los datos reducidos.

- **k-means** es un algoritmo de aprendizaje no supervisado que agrupa datos en clústeres. Selecciona puntos representativos llamados centroides y asigna cada punto al clúster más cercano en función de la distancia. Los centroides se actualizan iterativamente hasta que la asignación de puntos a clústeres sea estable. Finalmente, k-means encuentra grupos similares dentro de un conjunto de datos.

- **MMG (Modelo de Mixturas Gaussianas)** es un algoritmo de aprendizaje no supervisado que utiliza múltiples distribuciones gaussianas para representar y agrupar datos. Permite identificar patrones y asignar puntos de datos a diferentes clústeres basados en probabilidades.

- **Random Forest** es un algoritmo de aprendizaje supervisado que usa múltiples árboles de decisión. Cada árbol se construye de manera independiente con características y muestras aleatorias. Durante la predicción, se combina la salida de los árboles para obtener la clase o valor final. Se combina árboles para predecir con mayor precisión en clasificación y regresión.

- **SVM (Support Vector Machine)** es un algoritmo de aprendizaje supervisado. Su objetivo principal es encontrar un hiperplano óptimo que separe los datos en diferentes clases. Utiliza un enfoque de "márgenes máximos", donde busca el hiperplano que maximiza la distancia entre los puntos de datos más cercanos de diferentes clases. SVM se destaca por su capacidad para manejar eficazmente conjuntos de datos de alta dimensionalidad y su robustez frente a datos ruidosos.

- Una **ANN (Artificial Neural Network)**, es un modelo de aprendizaje automático. Consiste en una red interconectada de unidades llamadas neuronas artificiales, que procesan y transmiten información. Cada neurona toma entradas, aplica una función de activación y produce una salida. Las neuronas se organizan en capas, incluyendo una capa de entrada, una o más capas ocultas y una capa de salida. Durante el entrenamiento, se ajustan los pesos y los sesgos de las conexiones entre las neuronas para minimizar el error entre las salidas deseadas y las predicciones de la red. Las ANN son conocidas por su capacidad para aprender relaciones complejas y realizar tareas como clasificación, regresión y reconocimiento de patrones.

2.1.6. Métricas de evaluación

Las métricas de evaluación a utilizar en los modelos supervisados será **F1-score, precision y recall**. Estas métricas se usarán debido a que brindan una comprensión completa del rendimiento general de los modelos al considerar diferentes aspectos de su capacidad de predicción a diferencia de la métrica accuracy que no es tan completa.

Para calcular las métricas anteriores se debe tener en cuenta la **matriz de confusión**, la cual muestra la relación entre las predicciones del modelo y las clases reales en un formato tabular, es una matriz que desglosa las predicciones en cuatro categorías:

- Verdaderos positivos (TP): Instancias que el modelo clasificó correctamente como positivas.

- Falsos positivos (FP): Instancias que el modelo clasificó incorrectamente como positivas cuando en realidad son negativas.
- Verdaderos negativos (TN): Instancias que el modelo clasificó correctamente como negativas.
- Falsos negativos (FN): Instancias que el modelo clasificó incorrectamente como negativas cuando en realidad son positivas.

Con las 4 categorías anteriores podemos calcular el precision, recall y f1-score

2.1.6.1. Precision

El precision mide la proporción de casos positivos clasificados correctamente en relación con todos los casos clasificados como positivos.

$$\text{Precision} = \frac{TP}{TP + FP}$$

2.1.6.2. Recall

Recall mide la proporción de casos positivos capturados correctamente en relación con todos los casos positivos presentes en los datos reales.

$$\text{Recall} = \frac{TP}{TP + FN}$$

2.1.6.3. F1-score

El F1-score es una medida que combina precision y recall en un solo valor, proporcionando un equilibrio entre ambas métricas. Es especialmente útil cuando hay un desequilibrio entre las clases positivas y negativas en los datos.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Al utilizar el F1-score junto con precision y recall, se obtiene una imagen más completa del rendimiento de los modelos.

2.1.7. Búsqueda de hiperparametros

La búsqueda de hiperparámetros es el proceso en el aprendizaje automático que consiste en encontrar la configuración óptima de los parámetros ajustables de un modelo. Estos parámetros, conocidos como hiperparámetros, determinan el comportamiento y el rendimiento del modelo. La búsqueda de hiperparámetros implica explorar exhaustivamente o de manera aleatoria diferentes combinaciones de valores para estos parámetros y evaluar su impacto en las métricas de evaluación del modelo.

2.1.7.1. Grid Search

El objetivo de grid search, es probar cada combinación posible dado los rangos de cada hiperparámetro del modelo. Estos rangos se definen de acuerdo al criterio del experto y, una vez definido los rangos, el grid search busca cuál es la mejor combinación que da el mejor desempeño para el modelo con respecto a la métrica que se eligió (f1 score, accuracy, precision, etc.). Esta técnica ya está definida en la librería de sklearn, el cuál también usa la técnica de validación cruzada con k iteraciones para escoger el mejor modelo.

Ventajas:

- Es exhaustivo y garantiza que todas las combinaciones se prueben.
- Puede ser útil cuando se tiene un conocimiento limitado sobre los hiperparámetros adecuados.

Desventajas:

- Puede ser muy costoso en términos de tiempo y recursos computacionales, especialmente cuando se tienen muchos hiperparámetros y valores para ajustar.
- No toma en cuenta la información de iteraciones anteriores y puede no ser eficiente en la exploración del espacio de búsqueda.

2.1.7.2. Random Search

El objetivo de Random Search es probar un conjunto aleatorio de combinaciones de hiperparámetros dentro de los rangos definidos para cada hiperparámetro del modelo. Estos rangos se especifican previamente, al igual que en Grid Search. Random Search evalúa el rendimiento del modelo para un número limitado de combinaciones aleatorias y busca encontrar la mejor combinación que optimice la métrica seleccionada. Al igual que Grid Search, suele utilizarse en combinación con validación cruzada.

Ventajas:

- Es más eficiente en términos de tiempo y recursos que Grid Search, ya que no evalúa todas las combinaciones posibles.
- Puede encontrar buenas soluciones de manera más rápida debido a la aleatoriedad.

Desventajas:

- No garantiza una exploración exhaustiva del espacio de búsqueda, ya que se basa en selecciones aleatorias.

2.1.7.3. Optimización Bayesiana

La Optimización Bayesiana es un enfoque probabilístico que busca encontrar la mejor combinación de hiperparámetros utilizando un modelo de probabilidad para la función objetivo (por ejemplo, el rendimiento del modelo). En lugar de explorar todas las combinaciones, utiliza información acumulada de iteraciones anteriores para dirigir la búsqueda hacia regiones prometedoras del espacio de hiperparámetros. A medida que se evalúan más combinaciones, el modelo de probabilidad se ajusta y refina para encontrar la configuración óptima.

Ventajas:

- Es altamente eficiente en términos de tiempo y recursos, ya que se enfoca en áreas prometedoras del espacio de búsqueda.
- Puede encontrar configuraciones óptimas con menos evaluaciones que Grid Search o Random Search.

Desventajas:

- Requiere una configuración inicial más compleja, ya que debes especificar distribuciones de probabilidad para los hiperparámetros.
- No es adecuado para todos los tipos de problemas y puede no funcionar bien si la relación entre hiperparámetros y rendimiento es altamente no lineal o compleja.

2.1.8. Técnicas de evaluación

2.1.8.1. Validación cruzada

La validación cruzada es una técnica de aprendizaje automático utilizada para evaluar el rendimiento y la generalización de un modelo predictivo o clasificador. El objetivo es obtener una estimación más precisa del desempeño del modelo al entrenarlo y evaluarlo en distintos conjuntos de datos para garantizar que el modelo es independiente de la partición de los datos de entrenamiento y prueba. Este proceso implica dividir el conjunto de datos original en k particiones para entrenar y evaluar, de las k particiones, se escoge uno para evaluar, y los restantes k - 1 particiones, se escogen para entrenamiento. Posteriormente, se escoge otro conjunto como prueba y los demás como entrenamiento, y así iterativamente hasta escoger cada partición como conjunto de evaluación, por lo que, se hacen k iteraciones para escoger cada subconjunto como conjunto de evaluación. Al finalizar las k iteraciones, se promedian los resultados obtenidos y se obtiene la desviación estándar para observar si los resultados de cada iteración son cercanos o lejanos.

2.1.8.2. Validación hold out

La validación hold out, es una técnica de evaluación de modelos utilizada en el aprendizaje automático. Es una forma para dividir el conjunto de datos en conjuntos de entrenamiento y prueba para evaluar el rendimiento del modelo. Normalmente se hacen particiones del conjunto de datos en

70 % en entrenamiento 30 % para evaluación, si el conjunto es pequeño. Para bases de datos grandes, se divide 90 % para entrenamiento y 10 % para evaluación. Posteriormente, se hacen k iteraciones en donde el conjunto de datos se dividirá en el entrenamiento y la evaluación aleatoriamente para evaluar las distintas métricas (f1, precision, recall, etc), y se procede a sacar el promedio de las métricas calculadas para evaluar el rendimiento del modelo.

2.1.9. Silhouette Score

El Silhouette Score es una métrica utilizada en el algoritmo K-means para evaluar la calidad de la partición y encontrar el número óptimo de clusters. Calcula la distancia media intra-cluster y la distancia media inter-cluster para cada muestra. El valor resultante, que varía entre -1 y 1, indica qué tan bien se agrupan los datos y cuán separados están los clusters. Al maximizar el Silhouette Score, se puede determinar el número de clusters que proporciona una buena separación y coherencia en los datos.

2.1.10. Método del codo

El método del codo es una técnica en la búsqueda de hiperparametros que ayuda a determinar el número óptimo de clusters en un conjunto de datos. Se basa en la suma de las distancias cuadradas de los puntos a sus centroides. El proceso implica ejecutar el algoritmo de clustering para diferentes valores de k y calcular la suma de las distancias cuadradas. Luego, se traza una gráfica y se identifica el punto donde se produce un cambio significativo en la pendiente, formando un çodo". Este punto indica el número recomendado de clusters, ya que representa un equilibrio entre la cohesión dentro de los clusters y la separación entre ellos.

2.1.11. Estrategia de selección de representantes en Clusterización

2.1.11.1. Estrategia del Centroide

El representante de cada clúster se calcula como el punto que minimiza la suma de las distancias cuadradas hacia todos los puntos asignados a ese clúster. Este punto se denomina centroide y representa la ubicación central del clúster en el espacio de características

2.1.11.2. Estrategia de la media

El representante de cada clúster se calcula como el promedio de todos los puntos de datos asignados a ese clúster. Es decir, se calcula el centroide del clúster tomando el promedio de las coordenadas de todos los puntos.

2.1.12. Estimador de máxima verosimilitud

El estimador de máxima verosimilitud es un método que se puede usar para la búsqueda de hiperparametros que determina el número óptimo de componentes en las mixturas gaussianas. Este

es un método estadístico usado para estimar los mejores hiperparámetros de una distribución probabilística maximizando su verosimilitud. Para esto, se grafican los modelos usando los números de clusters en orden ascendente desde 2 hasta la cantidad de datos, luego, se observa en qué momento la gráfica tiende a ser constante y de este modo elegir el mejor número de clusters que maximice la verosimilitud.

2.1.13. Akaike information criterion (AIC)

Conocido en español como Criterio de Información de Akaike, el AIC es una medida utilizada en los modelos de mezclas gaussianas para la selección del modelo con el mejor número de clusters. El AIC se utiliza para comparar diferentes modelo y determinar cuál de ellos se ajusta mejor a los datos. El AIC penaliza los modelos más complejos al tener en cuenta tanto el ajuste a los datos como la complejidad del modelo, refiriéndose a la complejidad del modelo como la cantidad de parámetros que necesita el modelo para describir los datos de manera adecuada. Entre más bajo sea el AIC, más ajustados estarán los datos al modelo.

2.1.14. Bayesian information criterion (BIC)

Conocido en español como Criterio de Información Bayesiano, el objetivo principal del BIC es encontrar un equilibrio entre el ajuste del modelo a los datos y la simplicidad del modelo, penalizando los modelos que son demasiado complejos. En el contexto de los modelos de mezclas gaussianas, el BIC se utiliza para seleccionar el número óptimo de componentes (clusters) en el modelo de mixturas gaussianas. El BIC busca minimizar el valor total de la métrica, lo que significa encontrar un equilibrio entre un buen ajuste a los datos y la simplicidad del modelo. Para seleccionar el número óptimo de componentes en un modelo de mixturas gaussianas, se calcula el BIC para diferentes valores de "p"(número de componentes) y se selecciona el valor que minimiza el BIC.

2.2. Trabajos relacionados

- Diagnostic Classification of Cases of Canine Leishmaniasis Using Machine Learning [3]
  Esta investigación utilizó técnicas de aprendizaje automático para clasificar y predecir casos de leishmaniasis visceral canina a través de exámenes físicos. En el trabajó lograron las siguientes métricas: un accuracy del 75 %.

- Construction of Feed Forward MultiLayer Perceptron Model For Genetic Dataset in Leishmaniasis Using Cognitive Computing [6]
  En este trabajo investigativo se entrenó una red neuronal para diagnosticar la leishmaniasis, se usó un corpus de datos genéticos y se utilizaron métricas como accuracy, loss y error. Lograron al final un accuracy de 85 %

- GWAS-based machine learning approach to predict duloxetine response in major depressive disorder [4]

En esta investigación se enfrenta el problema de la selección de medicamentos mediante prueba y error para contra-restar la depresión, los investigadores mediante el uso de modelos de aprendizaje automático buscaron predecir qué desenlace terapéutico tendría un paciente con el medicamento duloxetine, con un corpus de diferentes genomas (GWAS) de pacientes, se entrenaron modelos de aprendizaje automático y lograron resultados de hasta un 66 % de accuracy.

Capítulo 3

Preprocesamiento de los datos

3.1. Dataset inicial

El dataset inicial consta de 72 filas, donde cada fila representa un paciente.
Se tienen 436 Columnas se divide así:

- Columna binaria (1, 0) representa el desenlace terapéutico
- Columna categórica con el Grupo Étnico al que pertenece el paciente, sus valores son 'Afrocolombiano' ú 'Otro'
- 434 columnas categóricas donde cada una representa un SNPs diferente en un Alelo especifico, estos 16 alelos representados con dos letras cada uno.

3.2. Transformaciones

3.2.1. Codificación

Como se dijo anteriormente las columnas de SNPs contienen letras, esto dificulta su procesamiento ya que los strings(letras) consumen mas recursos y dificultan el aprendizaje no supervisado. Por esta razón, se toma la decisión de realizar la transformación de codificar cada Alelo a un numero como se puede observar en el cuadro 3.1:

Cuadro 3.1: Codificación alelos

Alelos	Codificación
AA	0
TT	1
CC	2
GG	3
GC	4
GT	5
GA	6
CG	7
CT	8
CA	9
TG	10
TC	11
TA	12
AG	13
AC	14
AT	15

3.2.2. Codificación binaria única

Se realizó una codificación binaria para la variable de Grupo_étnico para facilitar el manejo de esta variable y tenerla en numérico, gracias a esto se remueve la columna Grupo_étnico y salen dos nuevas columnas 'Otro' y 'Afrocolombiano'. Esta variable es importante porque de acuerdo a las investigaciones, se ha observado que las personas de ascendencia afrodescendiente, tienen una mayor predisposición genética a la leishmaniasis.

3.3. Asignación de valores faltantes

El dataset utilizado presentaba valores nulos en las columnas de SNPs. Estos valores faltantes pueden afectar la calidad y validez de los análisis posteriores, particularmente en el caso del aprendizaje no supervisado.

Para abordar este problema, se implementó la asignación de valores faltantes utilizando la moda como estrategia. La moda se refiere al valor más frecuente en un conjunto de datos. Primero, se identifica el fenotipo de cada registro, que puede ser çura"(1) o "falla"(0), columna que representa el desenlace terapéutico. Luego, para cada columna que representa un SNP, se calcula el valor más común (moda) de esa columna específica, pero solo entre los registros que comparten el mismo fenotipo. Cuando se encuentra un valor nulo en una celda, se reemplaza por la moda correspondiente al fenotipo de ese registro. De esta manera, se aseguró que los valores asignados reflejaran las características más frecuentes y representativas de cada Fenotipo (cura ó falla) en particular.

Capítulo 4

Exploración de subconjuntos de SNPs

En la exploración de subconjuntos se busca encontrar los SNPs más significativos y dividirlos en subconjuntos para más tarde ser evaluados. Esta división se conoce como clusterización.

Los pasos a seguir por cada modelo son:

1. Preprocesar los datos
2. Buscar numero de clusters óptimo para el modelo y dataset
3. Realizar el entrenamiento del modelo y asignación de clusters
4. Seleccionar representantes de cada cluster utilizando las estrategias de selección

Los representantes de cada cluster de los modelos y utilizando una estrategia de selección diferente serán los subconjuntos que se utilizaran para evaluar.

4.1. Modelos de clusterización

Los modelos de clusterización a utilizar son:

- kmeans
- Gaussian mixture model (GMM)

4.1.1. Estrategias de selección de representantes

- Media
- Centroide

4.2. Preprocesamiento

Debido a que el objetivo es lograr encontrar subconjuntos de SNPs que sean significativos, se debe realizar una transposición del dataset, para que los SNPs queden como filas y se puedan agrupar utilizando las técnicas de clusters.

4.3. Clusterización

4.3.1. Kmeans

4.3.1.1. Búsqueda de numero de clusters

Se realizó la búsqueda del numero de clusters utilizando las siguientes técnicas con sus resultados:

- **Silhouette score**, donde según los resultados de la gráfica 4.1 se escogió **8 clusters**, recordemos que el rango de puntuación varía de -1 a 1, donde valores más altos indican una mejor separación entre clusters, por lo que se selecciona el primer pico mas alto.

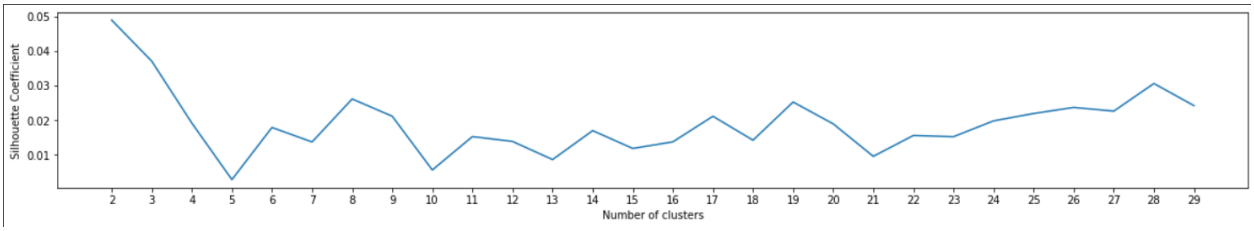

Figura 4.1: Resultados Silhouette score para Kmeans

- **Método del codo**, donde según los resultados de la gráfica 4.2 se escogió **5 clusters**, recordemos que la idea es encontrar el punto en la gráfica donde la inercia comienza a disminuir lentamente, lo que se asemeja a un codo en la curva. Este punto se considera un buen candidato para el número óptimo de clusters.

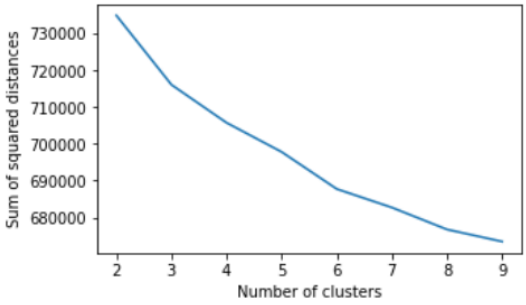

Figura 4.2: Resultados Método del codo para Kmeans

4.3.1.2. Selección de representantes

Después de calcular el numero óptimo de clusters en la sección anterior, se procede a realizar el entrenamiento y la asignación de clusters al dataset.

Una vez obtenidos los clusters, utilizaremos las estrategias de Media y centroides para encontrar los representantes de cada cluster.

Con las estrategias se seleccionaron 4 subconjuntos que se pueden observar en la tabla 4.1:

Cuadro 4.1: Subconjuntos seleccionados con Kmeans

Subconjunto	Estrategia	# Clusters	SNPs
0	Media	5	[rs6623659, rs12283577, rs2454566, rs9360356, ...]
1	Centroide	5	[rs6623659, rs9360356, rs2454566, rs4434842, ...]
2	Media	8	[rs12917202, rs10920017, rs9360356, rs4828978, ...]
3	Centroide	8	[rs12917202, rs10920017, rs9360356, rs4828978, ...]

4.3.2. GMM

4.3.2.1. Búsqueda de número de clusters

Se hizo la búsqueda del número de clusters usando varias técnicas, las cuales son:

- **Silhouette Score**, donde según los resultados de la gráfica 4.3 se escogen **363 clusters**, al ser el pico mas alto de clusters.

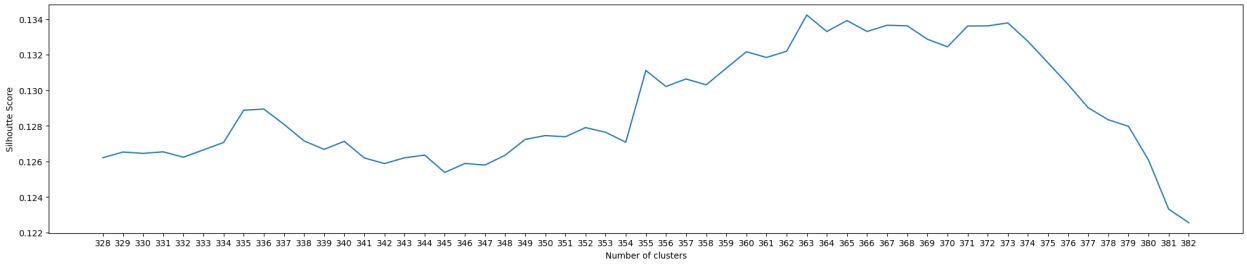

Figura 4.3: Resultados de Silhouette Score para GMM

- **Estimador de máxima verosimilitud**, donde según los resultados de la gráfica 4.4 se escogen **66 clusters**, recordemos que la idea es encontrar el punto en la gráfica donde tiende a ser constante.

- **AIC Score**, donde según los resultados de la gráfica 4.5 se escogen **21 clusters**, recordemos que entre mas bajo sea el AIC, mas ajustados estarán los datos al modelo, por lo tanto se busca un mínimo.

- **BIC Score**, donde según los resultados de la gráfica 4.6 se escogen **13 clusters**, recordemos que el objetivo es minimizar el BIC.

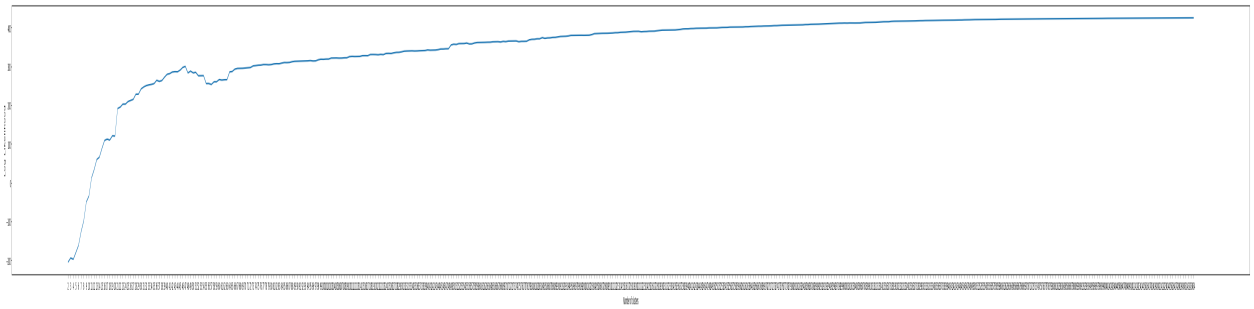

Figura 4.4: Resultados del estimador de máxima verosimilitud para GMM

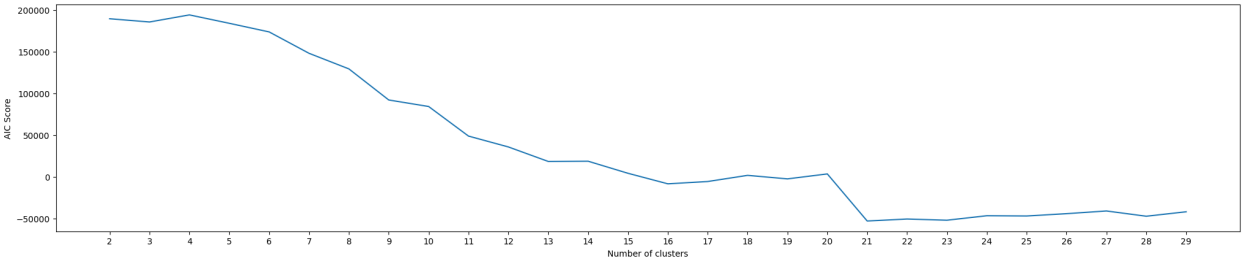

Figura 4.5: Resultados de AIC Score para GMM

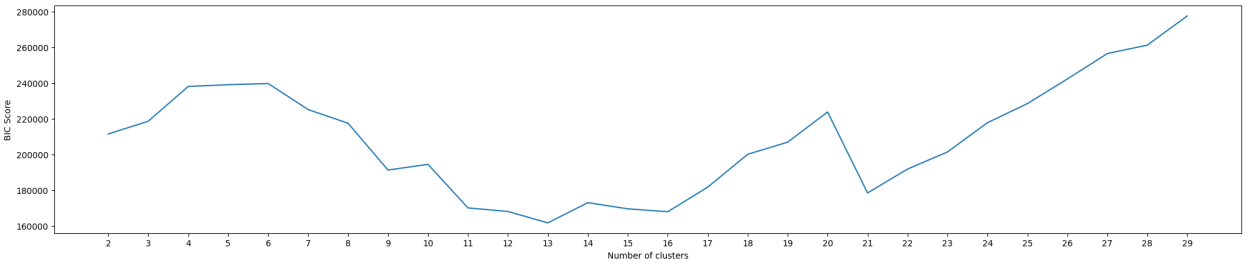

Figura 4.6: Resultados de BIC Score para GMM

4.3.2.2. Selección de representantes

Después de calcular el numero óptimo de clusters para GMM en la sección anterior, se procede a realizar el entrenamiento y la asignación de clusters al dataset. En GMM, en lugar de utilizar la estrategia de centroide como en K-Means, solo se emplea la estrategia de la media para encontrar los representantes de cada cluster, ya que GMM modela clusters como distribuciones gaussianas, y la media de estas distribuciones representa el centro del cluster. Esto difiere de K-Means, donde el centroide es un punto específico en el espacio de características.

El resultado de la selección de representantes se puede observar en la tabla 4.2

Cuadro 4.2: Subconjuntos encontrados con GMM

Subconjunto	# Clusters	SNPs
0	13	[rs10210872, rs2967255, rs6623659, rs10114675,...]
1	66	[rs17162075, rs2967255, rs10498844, rs4397348,...]
2	363	[rs17162075, rs2967255, rs11120748, rs12346493...]
3	21	[rs17162075, rs2967255, rs10498844, rs11962721...]

4.4. Subconjuntos finales

En la tabla 4.3 podemos observar todos los subconjuntos generados mediante clusterización utilizando los modelos de Kmeans y GMM en conjunto con las diferentes estrategias de selección de representantes.

Cuadro 4.3: Combinación de Subconjuntos encontrados con GMM y Kmeans

Subconjunto	# Clusters	SNPs
0	13	[rs10210872, rs2967255, rs6623659, rs10114675,...]
1	66	[rs17162075, rs2967255, rs10498844, rs4397348,...]
2	363	[rs17162075, rs2967255, rs11120748, rs12346493...]
3	21	[rs17162075, rs2967255, rs10498844, rs11962721...]
4	5	[rs6623659, rs12283577, rs2454566, rs9360356, ...]
5	5	[rs6623659, rs9360356, rs2454566, rs4434842, ...]
6	8	[rs12917202, rs10920017, rs9360356, rs4828978, ...]
7	8	[rs12917202, rs10920017, rs9360356, rs4828978, ...]

Capítulo 5

Evaluación de los subconjuntos de SNPs

Una vez identificado los subconjuntos, se debe evaluar el rendimiento de cada uno de estos. Para la evaluación, los subconjuntos se usan como entrada para el entrenamiento de modelos de aprendizaje supervisado. Y se considerará que los subconjuntos que permitan una predicción del desenlace terapéutico más ajustada a la realidad serán los mejores.

El proceso que se realiza es el siguiente:

1. Entrenar los modelos con un subconjunto arbitrario.
2. Realizar dos iteraciones de entrenamiento para búsqueda de hiperparametros, se escogen los hiperparametros de la iteración con mejores métricas para cada modelo.
3. Evaluar todos los subconjuntos en cada modelo (entrenado usando los hiperparametros anteriormente encontrados).
4. Escoger el subconjunto con mejores resultados en todos los modelos.
5. Entrenar los modelos con el subconjunto encontrado en el paso anterior.
6. Realizar dos iteraciones de entrenamiento para búsqueda de hiperparametros, se escogen los hiperparametros de la iteración con mejores métricas para cada modelo.
7. Evaluar todos los subconjuntos en cada modelo (entrenado usando los hiperparametros anteriormente encontrados) para su posterior análisis.

5.1. Técnicas de evaluación

Los modelos de aprendizaje supervisado utilizados son:

- Random Forest
- Support Vector Machine (SVM)
- Artificial Neural Network (ANN)

5.2. Entrenamiento con subconjunto 0

Para buscar los mejores resultados en la evaluación de los subconjuntos de SNPs, se realizó un ajuste de hiperparámetros utilizando grid search para encontrar los mejores valores de cada hiperparámetro en cada modelo. Para el primer entrenamiento se escoge el subconjunto 0 (ver en tabla 4.3) arbitrariamente, el subconjunto seleccionado podría no ser el que de mejores resultados, por lo que para no afectar el proceso, se volverá a hacer un entrenamiento de los modelos con el subconjunto que mejor resultados genere en la primera ronda de evaluación.

5.2.1. Iteración 1

En la primera iteración, se buscan los rangos de cada uno de los parámetros de cada modelo, el rango de valores estimados se muestran en las tablas 5.1, 5.2 y 5.3.

Cuadro 5.1: Grilla de valores a probar para la búsqueda de hiperparámetros del modelo Random Forest

Random Forest	
Parámetro	Rango de valores
n estimators	25, 50, 75, 100, 125, 150, 175, 200
max depth	None, 2, 4, 6, 8, 10, 12, 14
min samples split	2, 4, 6, 8, 10, 12, 14, 16
min samples leaf	2, 4, 6, 8, 10, 12, 14, 16

Cuadro 5.2: Grilla de valores a probar para la búsqueda de hiperparámetros del modelo Support Vector Machine

Support Vector Machine	
Parámetro	Rango de valores
c	0.001, 0.01, 0.05, 0.1, 1.0, 2.0, 2.5, 3.0, 3.5, 4.0, 10.0
gamma	'scale', 'auto', 2, 4, 6, 8, 10, 12, 14
kernel	'linear', 'poly', 'rbf', 'sigmoid'

Con los rangos definidos para cada hiperparámetro de cada modelo, se hace el proceso de grid search, y se escoge el mejor valor, esto se hace mediante el proceso de validación cruzada utilizando 5 bloques (folds).

- Los mejores valores de hiperparametros para **Random Forest** se muestran en la tabla 5.4.
- Los mejores valores de hiperparametros para **Support Vector Machine** se muestran en la tabla 5.5.

Cuadro 5.3: Grilla de valores a probar para la búsqueda de hiperparámetros del modelo Artificial Neural Network

Artificial Neural Network	
Parámetro	Rango de valores
learning rate	0.00001, 0.0001, 0.001, 0.003, 0.03, 0.1, 1
num layers	1, 2, 3, 4
num neurons	10, 30, 50
batch size	8, 16, 32

- Los mejores valores de hiperparametros para **Artificial Neural Network** se muestran en la tabla 5.6.

Cuadro 5.4: Random Forest mejores valores hiperparámetros

Random Forest	
Parámetro	Mejor valor
n estimators	25
max depth	2
min samples split	4
min samples leaf	2

Cuadro 5.5: Support Vector Machine mejores valores hiperparámetros

Support Vector Machine	
Parámetro	Mejor valor
c	0.01
gamma	0.001
kernel	linear

Cuadro 5.6: Artificial Neural Network mejores valores hiperparámetros

Artificial Neural Network	
Parámetro	Mejor valor
learning rate	0.001
num layers	3
num neurons	30
batch size	8

5.2.2. Iteración 2

Para la segunda iteración, se definen los rangos de cada uno de los parámetros de cada modelo a partir de los mejores valores encontrados para los parámetros en la primera iteración, es decir, nuevos rangos que estén alrededor de los mejores valores que se encontraron en la iteración 1. Esto con el fin de posiblemente encontrar mejores valores para los parámetros y que de un mejor resultado. Con base en esto, los rangos definidos se muestran en las tablas 5.7, 5.8 y 5.9

Cuadro 5.7: Random Forest Hiperparámetros

Random Forest	
Parámetro	Rango de valores
n estimators	15, 30, 45, 60, 75, 90
max depth	None, 3
min samples split	3, 5
min samples leaf	3

Cuadro 5.8: Support Vector Machine Hiperparámetros

Support Vector Machine	
Parámetro	Rango de valores
c	0.005, 0.01, 0.015, 0.02, 0.025, 0.03
gamma	0.001, 0.0005, 0.0015, 0.002, 0.0025, 0.003

Igual que en la iteración 1, se hace el proceso de grid search sobre los rangos obtenidos, y se escoge el nuevo mejor valor usando validación cruzada.

- Los mejores valores de hiperparametros para **Random Forest** en la segunda iteración se muestran en la tabla 5.10.

Cuadro 5.9: Artificial Neural Network Hiperparámetros

Artificial Neural Network	
Parámetro	Rango de valores
learning rate	0.0005, 0.008, 0.002
num layers	1, 2
num neurons	25, 35
batch size	2, 4

- Los mejores valores de hiperparametros para **Support Vector Machine** en la segunda iteración se muestran en la tabla 5.11.
- Los mejores valores de hiperparametros para **Artificial Neural Network** en la segunda iteración se muestran en la tabla 5.12.

Cuadro 5.10: Random Forest mejores valores hiperparámetros

Random Forest	
Parámetro	Mejor valor
n estimators	30
max depth	None
min samples split	3
min samples leaf	3

Cuadro 5.11: Support Vector Machine mejores valores hiperparámetros

Support Vector Machine	
Parámetro	Mejor valor
c	0.01
gamma	0.001
kernel	linear

Cuadro 5.12: Artificial Neural Network mejores valores hiperparámetros

Artificial Neural Network	
Parámetro	Mejor valor
learning rate	0.0005
num layers	2
num neurons	35
batch size	4

5.3. Evaluación de los subconjuntos

Una vez definidos los valores de los hiperparámetros para cada modelo de aprendizaje supervisado, se procedió a evaluar cada subconjunto utilizando la técnica de validación hold-out en 15 iteraciones. Para cada modelo, se registró la media de las 15 iteraciones junto con la desviación estándar (Representada con σ) correspondiente para cada métrica. Los resultados para cada modelo se muestran en las tablas 5.13, 5.14 y 5.15

Cuadro 5.13: Random Forest Resultados

Random Forest						
Subconjunto	F1	σ de F1	Precision	σ de Precision	Recall	σ de Recall
0	0.848267	0.065049	0.801845	0.125700	0.920524	0.062503
1	0.844720	0.065209	0.780150	0.119050	0.937941	0.054150
2	**0.864636**	**0.048872**	**0.831993**	**0.085359**	**0.910655**	**0.074342**
3	0.854550	0.059130	0.804961	0.085705	0.922619	0.079663
4	0.797029	0.054798	0.753527	0.086228	0.857863	0.078072
5	0.794672	0.064840	0.740711	0.105481	0.875979	0.097821
6	0.840703	0.040195	0.785729	0.078492	0.917050	0.078608
7	0.818699	0.053475	0.748865	0.064969	0.912885	0.090945

Cuadro 5.14: Support Vector Machine Resultados

Support Vector Machine						
Subconjunto	F1	σ de F1	Precision	σ de Precision	Recall	σ de Recall
0	0.807818	0.076600	0.755812	0.124325	0.882583	0.069216
1	0.815516	0.071450	0.747550	0.140908	0.925104	0.075527
2	**0.879260**	**0.051013**	**0.843616**	**0.107793**	**0.933466**	**0.075063**
3	0.829820	0.064093	0.780440	0.108975	0.901403	0.069125
4	0.771837	0.060311	0.707679	0.082431	0.866847	0.110484
5	0.827060	0.072479	0.740685	0.077318	0.945106	0.051988
6	0.826734	0.042553	0.754764	0.086526	0.922825	0.085485
7	0.803637	0.055556	0.731869	0.080124	0.901807	0.082788

Cuadro 5.15: Artificial Neural Network Resultados

Artificial Neural Network						
Subconjunto	F1	σ de F1	Precision	σ de Precision	Recall	σ de Recall
0	0.747271	0.072598	0.773509	0.120698	0.744890	0.111089
1	0.730599	0.071264	0.726294	0.090422	0.754096	0.115638
2	**0.822634**	**0.090455**	**0.808336**	**0.112108**	**0.848987**	**0.100456**
3	0.770989	0.092703	0.776648	0.114891	0.787629	0.139768
4	0.703039	0.078535	0.720493	0.088898	0.709352	0.134963
5	0.762615	0.089264	0.707609	0.093206	0.833082	0.102824
6	0.756734	0.058740	0.727007	0.106017	0.812836	0.113667
7	0.707473	0.093200	0.666492	0.131084	0.779058	0.106978

5.3.1. Análisis de Resultados de Random Forest

Se realiza un análisis en el modelo de Random Forest por cada métrica evaluada. Los valores se pueden observar en la tabla 5.13.

5.3.1.1. F1 Score

- El subconjunto 2 tiene el f1 score más alto (0.864636) y la desviación estándar más baja (0.048872), lo que indica un rendimiento consistente y sólido.
- Los subconjuntos 0, 1, 3, 6 y 7 también tienen buenos valores de f1 score, aunque ligeramente más bajos que el subconjunto 2.
- Los subconjuntos 4 y 5 tienen los valores más bajos de f1 score, lo que sugiere un rendimiento deficiente en comparación con otros subconjuntos.

5.3.1.2. Precision

- Nuevamente, el subconjunto 2 muestra el mejor precision (0.831993) con una baja desviación estándar (0.085359), indicando que es capaz de identificar positivos verdaderos con precision.
- Los subconjuntos 0, 1, 3 y 6 también tienen valores de precision decentes, aunque ligeramente más bajos que el subconjunto 2.
- Los subconjuntos 4 y 5 tienen el precision más bajo, lo que sugiere un mayor número de falsos positivos.

5.3.1.3. Recall

- El subconjunto 1 muestra el recall más alto (0.937941) con una desviación estándar baja (0.054150), lo que indica que es capaz de identificar la mayoría de los positivos verdaderos.
- El subconjunto 0 también tiene un buen valor de recall.
- El subconjunto 2, a pesar de tener un f1 score y precision sólidos, tiene un recall ligeramente más bajo, lo que sugiere que podría estar perdiendo algunos positivos verdaderos.
- Los subconjuntos 4 y 5 tienen los valores de recall más bajos, lo que indica que no identifican positivos verdaderos de manera efectiva.

En resumen, el subconjunto 2 parece ser el mejor en términos generales, con buenos valores de f1 score y precision, aunque su recall es un poco más bajo. El subconjunto 1 también destaca en términos de recall. Los subconjuntos 4 y 5 tienen el peor rendimiento en todas las métricas.

5.3.2. Análisis de Resultados de SVM

A continuación, se realiza un análisis de los resultados de la tabla de Support Vector Machine (SVM) por cada métrica y se destacan los subconjuntos que se desempeñan mejor o peor en cada métrica. Los valores se pueden observar en la tabla 5.14.

5.3.2.1. F1 Score

- El subconjunto 2 tiene el f1 score más alto (0.879260) y la desviación estándar más baja (0.051013), lo que indica un rendimiento consistente y sólido en términos de equilibrio entre precision y exhaustividad.
- Los subconjuntos 1 y 5 también tienen buenos valores de f1 score, aunque ligeramente más bajos que el subconjunto 2.
- Los subconjuntos 0, 3, 6 y 7 tienen la métrica de f1 score decentes, pero en general, sus valores son más bajos que el subconjunto 2.
- El subconjunto 4 tiene el f1 score más bajo, sugiriendo el peor rendimiento en esta métrica.

5.3.2.2. Precision

- El subconjunto 2 muestra la mejor precision (0.843616) con una desviación estándar relativamente baja (0.107793), indicando que es capaz de identificar positivos verdaderos con precision.
- El subconjunto 5 también tiene una buena precision.
- Los subconjuntos 0, 1, 3, 6 y 7 tienen precision decente, pero en general, sus valores son más bajos que el subconjunto 2.
- El subconjunto 4 tiene la precision más baja, lo que sugiere que tiene un alto número de falsos positivos.

5.3.2.3. Recall

- El subconjunto 2 muestra un buen recall (0.933466) con una desviación estándar baja (0.075063), lo que indica que es capaz de identificar la mayoría de los positivos verdaderos.
- El subconjunto 5 tiene el recall más alto, lo que sugiere que es excelente para identificar positivos verdaderos, pero podría tener un mayor número de falsos negativos en comparación con el subconjunto 2.
- Los subconjuntos 0, 1, 3, 6 y 7 también tienen valores de recall decentes, aunque ligeramente más bajos que el subconjunto 2.
- El subconjunto 4 tiene el recall más bajo, lo que indica que no identifica positivos verdaderos de manera efectiva.

En resumen, el subconjunto 2 destaca en términos generales en todas las métricas. El subconjunto 5 también muestra un buen desempeño en términos de recall y precision, pero su f1 score es ligeramente inferior al del subconjunto 2. El subconjunto 4 tiene el peor rendimiento en todas las métricas.

5.3.3. Análisis de Resultados de ANN

A continuación, se realiza un análisis de los resultados de la tabla de Artificial Neural Network (ANN) por cada métrica y se destacan los subconjuntos que se desempeñan mejor o peor en cada métrica. Los valores se pueden observar en la tabla 5.15.

5.3.3.1. F1 Score

- El subconjunto 2 tiene el f1 score más alto (0.822634) y la desviación estándar más baja (0.090455), lo que indica un rendimiento consistente.

- Los subconjuntos 5 y 6 también tienen buenos valores de f1 score, aunque ligeramente más bajos que el subconjunto 2.

- Los subconjuntos 0, 1 y 3 tienen valores de f1 score decentes, pero en general, sus valores son más bajos que el subconjunto 2.

- Los subconjuntos 4 y 7 tienen los valores de f1 score más bajos.

5.3.3.2. Precision

- El subconjunto 2 muestra la mejor precision (0.808336) con una desviación estándar relativamente baja (0.112108), indicando que es capaz de identificar positivos verdaderos con precision.

- El subconjunto 5 también tiene una buena precision.

- Los subconjuntos 0, 3, 6 y 7 tienen precision decente, pero en general, sus valores son más bajos que el subconjunto 2.

- Los subconjuntos 1 y 4 tienen las Precisiones más bajas, lo que sugiere un alto número de falsos positivos.

5.3.3.3. Recall

- El subconjunto 2 muestra un buen recall (0.848987) con una desviación estándar baja (0.100456), lo que indica que es capaz de identificar la mayoría de los positivos verdaderos.

- El subconjunto 5 tiene el recall más alto, lo que sugiere que es excelente para identificar positivos verdaderos, pero podría tener un mayor número de falsos negativos en comparación con el subconjunto 2.

- Los subconjuntos 0, 3, 6 y 7 también tienen valores de recall decentes, aunque ligeramente más bajos que el subconjunto 2.

- Los subconjuntos 1 y 4 tienen los valores del recall más bajos, lo que indica que no identifican positivos verdaderos de manera efectiva.

En resumen, el subconjunto 2 destaca en términos generales en todas las métricas, con el f1 score y la precision más altos, y un recall sólido. El subconjunto 5 también muestra un buen desempeño en términos de recall y precision, pero su f1 score es ligeramente inferior al del subconjunto 2. Los subconjuntos 1 y 4 tienen el peor rendimiento en la mayoría de las métricas.

5.3.4. Conclusión análisis de modelos

El **subconjunto 2** se destaca como mejor en términos de rendimiento general, porque obtuvo los valores más altos en la mayoría de métricas en comparación con otros subconjuntos. Además, la desviación estándar fue relativamente baja, lo que significa que tiene un rendimiento consistente en los modelos. Aunque se debería dar mas importancia al recall dado que el problema es identificar las características de los pacientes que no se logran curar para evitar tratamientos médicos con consecuencias negativas, se opta por darle igual relevancia a las tres métricas teniendo en cuenta que se realizará una mejora en el entrenamiento. Como resultado, se elige el subconjunto 2 para el siguiente paso (ver Sección 5.4).

Los subconjuntos 5 y 6 también mostraron un rendimiento sólido en varias métricas en los tres modelos, aunque no superaron al subconjunto 2. Estos subconjuntos destacan porque tienen buenas métricas y tienen una cantidad de SNPs considerablemente menor que el subconjunto 2 (ver cuadro 4.3), esto es bueno debido a buscamos reducir la dimensionalidad en el dataset de predicción.

En contraste, los subconjuntos 0, 1, 3, 4 y 7 generalmente mostraron un rendimiento inferior en comparación con los subconjuntos mencionados anteriormente.

5.4. Mejora en el entrenamiento con nuevo subconjunto

Se usó el subconjunto 0 (elegido arbitrariamente) para definir los mejores valores que se usarían para los parámetros de los modelos, ahora con los resultados obtenidos en las tablas de la sección 5.3, se realiza la mejora en el entrenamiento con el subconjunto 2.

5.4.1. Iteración 1

Para la primera iteración, repetimos el procedimiento. Se buscan los rangos de cada uno de los parámetros, pero, estos rangos siguen siendo los mismos ya que los rangos que se encontraron aplican para cualquiera de los subconjuntos de datos. Por lo que, las tablas de los rangos encontrados para los parámetros están en las tablas de la sección 5.2.1. Los resultados de realizar el proceso de grid search sobre las tablas, usando como datos de entrenamiento el subconjunto 2 .

- Los mejores valores de hiperparametros para **Random Forest** en la mejora de entrenamiento con el subconjunto 2 se muestran en la tabla 5.16.
- Los mejores valores de hiperparametros para **Support Vector Machine** en la mejora de entrenamiento con el subconjunto 2 se muestran en la tabla 5.17.
- Los mejores valores de hiperparametros para **Artificial Neural Network** en la mejora de entrenamiento con el subconjunto 2 se muestran en la tabla 5.18.

Se puede observar que el rendimiento en el entrenamiento es mejor cuando se usó el subconjunto 2 que cuando se usó el subconjunto 0. Por lo que, se obtuvo mejores valores para los parámetros de los modelos, se hará una segunda iteración para tratar de mejorar aun mas las métricas.

Cuadro 5.16: Random Forest mejores valores hiperparámetros

Random Forest	
Parámetro	Mejor valor
n estimators	100
max depth	10
min samples split	6
min samples leaf	2

Cuadro 5.17: Support Vector Machine mejores valores hiperparámetros

Support Vector Machine	
Parámetro	Mejor valor
c	1.0
gamma	scale
kernel	rbf

Cuadro 5.18: Artificial Neural Network mejores valores hiperparámetros

Artificial Neural Network	
Parámetro	Mejor valor
learning rate	0.003
num layers	4
num neurons	30
batch size	32

5.4.2. Iteración 2

Para la iteración 2, definimos nuevamente los rangos, y que estos estén alrededor de los mejores valores obtenidos en las tablas de la iteración 1, de esta forma, el rendimiento nos debe dar mejor o igual que el de la iteración 1. Los nuevos rangos a considerar de los parámetros de cada modelo se pueden ver en las tablas 5.19, 5.20 y 5.21

Cuadro 5.19: Random Forest hiperparametros

Random Forest	
Parámetro	Rango de valores
n estimators	25, 50, 75, 100, 125, 150, 175, 200
max depth	9, 11
min samples split	5, 7
min samples leaf	1, 3

Cuadro 5.20: Support Vector Machine hiperparametros

Support Vector Machine	
Parámetro	Rango de valores
c	0.5, 0.6, 0.7, 0.8, 0.7, 0.9, 1.1, 1.2, 1.3, 1.4, 1.5

Cuadro 5.21: Artificial Neural Network hiperparametros

Artificial Neural Network	
Parámetro	Rango de valores
learning rate	0.002, 0.008, 0.01
num layers	5, 6, 7
num neurons	25, 30, 35
batch size	64

Con los nuevos rangos, se entrenaron los modelos para sacar los mejores valores para los parámetros usando grid search y validación cruzada. Los mejores valores se dividen así:

- Los mejores valores de hiperparametros para **Random Forest** en la segunda iteración en la mejora de entrenamiento con el subconjunto 2 se muestran en la tabla 5.22.
- Los mejores valores de hiperparametros para **Support Vector Machine** en la segunda iteración en la mejora de entrenamiento con el subconjunto 2 se muestran en la tabla 5.23.
- Los mejores valores de hiperparametros para **Artificial Neural Network** en la segunda iteración en la mejora de entrenamiento con el subconjunto 2 se muestran en la tabla 5.24.

Cuadro 5.22: Random Forest mejores valores hiperparámetros

Random Forest	
Parámetro	Mejor valor
n estimators	95
max depth	9
min samples split	7
min samples leaf	3

Cuadro 5.23: Support Vector Machine mejores valores hiperparámetros

Support Vector Machine	
Parámetro	Mejor valor
c	1.1
gamma	scale
kernel	rbf

Cuadro 5.24: Artificial Neural Network mejores valores hiperparámetross

Artificial Neural Network	
Parámetro	Mejor valor
learning rate	0.008
num layers	6
num neurons	35
batch size	64

5.5. Evaluación de los subconjuntos con los nuevos parámetros

Una vez obtenido los nuevos hiperparámetros de cada modelo, se procedió a evaluar cada subconjunto utilizando la técnica de validación hold-out en 15 iteraciones. Para cada modelo, se registró la media de las 15 iteraciones junto con la desviación estándar (Representada con σ) correspondiente para cada métrica. Los resultados para cada modelo se muestran en las tablas 5.25, 5.26 y 5.27. El análisis de estos resultados se realiza en el capitulo 6.

Cuadro 5.25: Random Forest Resultados

Random Forest						
Subconjunto	F1	σ de F1	Precision	σ de Precision	Recall	σ de Recall
0	0.848267	0.065049	0.801845	0.125700	0.920524	0.062503
1	0.844720	0.065209	0.780150	0.119050	0.937941	0.054150
2	0.864636	0.048872	0.831993	0.085359	0.910655	0.074342
3	0.854550	0.059130	0.804961	0.085705	0.922619	0.079663
4	0.797029	0.054798	0.753527	0.086228	0.857863	0.078072
5	0.794672	0.064840	0.740711	0.105481	0.875979	0.097821
6	0.840703	0.040195	0.785729	0.078492	0.917050	0.078608
7	0.818699	0.053475	0.748865	0.064969	0.912885	0.090945

Cuadro 5.26: Support Vector Machine Resultados

Support Vector Machine						
Subconjunto	F1	σ de F1	Precision	σ de Precision	Recall	σ de Recall
0	0.807818	0.076600	0.755812	0.124325	0.882583	0.069216
1	0.815516	0.071450	0.747550	0.140908	0.925104	0.075527
2	0.879260	0.051013	0.843616	0.107793	0.933466	0.075063
3	0.829820	0.064093	0.780440	0.108975	0.901403	0.069125
4	0.771837	0.060311	0.707679	0.082431	0.866847	0.110484
5	0.827060	0.072479	0.740685	0.077318	0.945106	0.051988
6	0.826734	0.042553	0.754764	0.086526	0.922825	0.085485
7	0.803637	0.055556	0.731869	0.080124	0.901807	0.082788

Cuadro 5.27: Artificial Neural Network Resultados

Artificial Neural Network						
Subconjunto	F1	σ de F1	Precision	σ de Precision	Recall	σ de Recall
0	0.739668	0.058369	0.753012	0.103441	0.745397	0.097488
1	0.763600	0.083015	0.784610	0.087745	0.755049	0.117998
2	0.841746	0.053205	0.828445	0.108592	0.874847	0.090239
3	0.777096	0.074611	0.729904	0.099422	0.859002	0.130405
4	0.714091	0.086860	0.739672	0.113483	0.735089	0.170042
5	0.784494	0.072313	0.744916	0.088709	0.838170	0.101984
6	0.776093	0.084126	0.776063	0.102790	0.798651	0.142277
7	0.719828	0.076192	0.650408	0.099887	0.819208	0.073950

Capítulo 6

Análisis de resultados

6.1. Subconjuntos representativos

Si bien se usaron varias técnicas para identificar SNPs representativos, se puede observar que varios SNPs se encuentran en distintos subconjuntos, podemos observar en la tabla 6.1 los SNPs que más se repiten y su gen asociado.

Cuadro 6.1: SNPs más repetitivos y su gen asociado

SNPs		
SNP	Gen asociado	Número de apariciones en subconjuntos distintos
rs9360356	ADGRB3: Variante del intrón	7
rs2454566	PRKG1: Variante del intrón	7
rs6623659	DACH2: Variante del intrón	5
rs2967255	LOC105371241 : Variante del intrón	4
rs14391	LOC105371483: Variante 500B Downstream	4
rs12917202	PYGO1: Variante del intrón	4
rs12037569	SORT1: Variante del intrón	4
rs10920017	CAMSAP2: Variante 2KB Upstream	3
rs11962721	PRKN : Variante del intrón	3
rs4828978	PTCHD1-AS: Variante del intrón	3
rs6045443	SEC23B: Variante del intrón	3
rs4855026	SOX2-OT: Variante del intrón	3
rs17162075	TMEM178B: Variante del intrón	3

Los SNPs repetitivos que se pueden apreciar en la tabla 6.1 demuestra la relación que tienen en el proceso para determinar la cura o falla del tratamiento de la leishmaniasis ya que la extracción de estos SNPs se hizo a partir de la maximización del recall para determinar si el paciente se debe someter o no al tratamiento de la leishmaniasis.

Se pueden observar las siguientes características de los SNPs de la tabla 6.1

- PRKG1 (rs2454566): El gen PRKG1 está relacionado con la señalización del óxido nítrico y

el GMP cíclico, que pueden estar vinculados a la función cardiovascular y posiblemente a la regulación de la respuesta inmune, aunque se requiere investigación adicional.

- SEC23B (rs6045443): SEC23B desempeña un papel en el transporte de proteínas intracelulares, lo que podría influir en procesos relacionados con la respuesta inmune y la inflamación.
- CAMSAP2 (rs10920017): Es un gen que está involucrado en vías de señalización de calcio y calmodulina, lo que puede tener un impacto indirecto en las respuestas inmunológicas. Aunque no está directamente relacionado con el sistema inmunológico, las vías de señalización que involucran al calcio son importantes para funciones celulares y, en algunos casos, pueden influir en la respuesta inmunológica.
- TMEM178B (rs17162075): Aunque la función exacta de TMEM178B no está bien caracterizada, los genes de membrana a menudo están involucrados en la interacción celular, que puede ser relevante en la respuesta inmunológica y otros procesos celulares.

Recordemos que la señalización celular permite que las células detecten y respondan a señales químicas específicas, como hormonas, neurotransmisores o factores de crecimiento, que actúan como mensajeros en el organismo. El resto de SNPs identificados en la tabla 6.1 tienen gran importancia ya que además de las características mencionadas, también contribuyen a la señalización celular, esto juega un papel muy importante en la regulación de la respuesta inmunitaria del cuerpo humano, por lo tanto, esto contribuye en la resolución de la infección de la leishmaniasis [1]. Estudiar estos SNPs ayuda para la predicción del desenlace terapéutico de la leishmaniasis.

6.2. Análisis y mejores modelos

- **Random Forest**
  - El modelo Random Forest muestra un rendimiento sólido en términos de f1, con valores que oscilan entre aproximadamente 0.794 y 0.864 en diferentes subconjuntos. Además, su desviación estándar de F1 es relativamente baja, lo que sugiere una consistencia en el rendimiento del modelo en diferentes subconjuntos.
  - La precision también es aceptable, con valores entre alrededor de 0.740 y 0.832. Lo que indica que cuando predice una clase positiva, tiende a ser correcto en aproximadamente el 77 % de las ocasiones.
  - La métrica recall es alta en la mayoría de los subconjuntos, lo que sugiere que el modelo tiene una buena capacidad para identificar positivos verdaderos.
  - Las desviaciones estándar en f1, precision y recall indican poca variabilidad en el rendimiento entre subconjuntos porque son bajas, oscilando entre los valores de [0.04, 0.06] para el f1, [0.06, 0.12] para el precision, y [0.05, 0.09] para el recall.

[1] Asesoría realizada por Maria Adelaida Gomez, PhD. Coordinadora de Biología Molecular y Bioquímica del CIDEIM, [27/07/2023].

- **Support Vector Machine**
  - El modelo SVM también muestra un rendimiento decente en F1, con valores entre aproximadamente 0.772 y 0.879 en diferentes subconjuntos. Al igual que Random Forest, la desviación estándar de F1 es baja, lo que indica consistencia en el rendimiento
  - El precision varía entre 0.707 y 0.844, lo que indica que es un poco menos preciso que Random Forest en la tarea.
  - La métrica recall es alta en varios subconjuntos, lo que sugiere que el modelo es efectivo para detectar positivos verdaderos.
  - Al igual que Random Forest, las desviaciones estándar en las métricas son moderadas.
- **Artificial Neural Network**
  - El modelo de Redes Neuronales Artificiales muestra resultados variables en F1, con valores que oscilan entre aproximadamente 0.714 y 0.842 en diferentes subconjuntos.
  - La precision varía entre 0.651 y 0.784, son valores bajos a comparación del resto de modelos.
  - La métrica recall es aceptable en varios subconjuntos, pero tiene un rendimiento inferior en la identificación de verdaderos positivos en comparación con los otros dos modelos.
  - Las desviaciones estándar en f1, precision y recall son relativamente altas en comparación con los otros dos modelos, lo que podría indicar una mayor variabilidad en el rendimiento.

Teniendo en cuenta lo anterior, el modelo de Random Forest muestra un rendimiento positivo y sólido en términos de las métricas f1, precision y recall, lo que significa que este modelo está realizando un buen trabajo en la tarea de clasificación o predicción, además, podemos resaltar la desviación estándar calculada, la cual es baja, y por tanto, se entiende que las métricas obtenidas no tienen mucha variabilidad al evaluar con distintos conjuntos. Adicionalmente, se puede resaltar la métrica del recall, ya que esta fue alta, y nos sugiere que el modelo tiene una buena capacidad para identificar positivos verdaderos, lo que significa, una respuesta más acertada para saber si un paciente se va a curar o no al someterse al tratamiento de la leishmaniasis. También se puede observar que el modelo de Support Vector Machine muestra buenos resultados en términos de f1, precision y recall, lo que indica, nuevamente, que el modelo está mostrando un buen desempeño en la tarea de predicción. Asimismo, al tener desviaciones estándar bajas, se entiende que el modelo se comporta similar independientemente del conjunto de evaluación. Además, debido a que el recall es alto, se puede asegurar que hace bien la tarea de predicción ante la respuesta de la cura del tratamiento de la leishmaniasis ya que esta métrica identifica los verdaderos casos positivos. Por otro lado, los resultados de las métricas del modelo de Artificial Neural Network son inferiores que los demás, además, la desviación estándar es considerablemente alta, por lo que este modelo se le dificulta la tarea de predicción y por tanto, se le dificulta una respuesta acertada para saber si un paciente se va a curar o no al someterse al tratamiento de la leishmaniasis.

Con base en las características de los resultados, se escoge el **Random Forest** como mejor modelo, en resumen destaca por lo siguiente:

- Equilibrio en f1 score: Random Forest logra un f1 score promedio más alto en comparación con los otros modelos.
- Alto recall
- Buen precision: Aunque el precision de Random Forest (alrededor de 0.77) es ligeramente inferior a su recall, sigue siendo sólida. Y por la importancia de descartar falsos positivos se tiene en cuenta el buen precision de Random Forest
- Consistencia: Random Forest muestra una baja desviación estándar en sus métricas, lo que sugiere consistencia en el rendimiento en diferentes subconjuntos. Esto es importante para la fiabilidad del modelo

Por otro lado, según los resultados, el subconjunto que tiene las mejores métricas fue el subconjunto 2, este resultado es esperado debido a que es el subconjunto que tiene más SNPs (363 SNPs), sin embargo, no mostró una gran diferencia de resultados con respecto a los demás subconjuntos encontrados, lo cual nos dice que se puede reducir la dimensionalidad de los subconjuntos utilizados en el subconjunto 2. Teniendo en cuenta la métrica de recall (la cual es la métrica de interés), podemos observar que los resultados del recall en los distintos subconjuntos son muy similares al resultado del subconjunto 2, de hecho, los subconjuntos 0, 1, 3, 6, y 7 tienen mejores resultados que el subconjunto 2, esto se puede ver en la tabla 6.2.

Cuadro 6.2: Resultado del recall de todos los subconjuntos usando el modelo de Random Forest

Random forest		
Subconjunto	Recall	Recall std
0	0.920524	0.062503
1	0.937941	0.054150
2	0.910655	0.074342
3	0.922619	0.079663
4	0.857863	0.078072
5	0.875979	0.097821
6	0.917050	0.078608
7	0.912885	0.090945

A pesar de que Random Forest fue el mejor modelo comparando las métricas con todos los subconjuntos encontrados, podemos observar que, con el subconjunto 2, el modelo de Support Vector Machine obtuvo un mejor rendimiento en las métricas al evaluar el subconjunto 2 solamente, esto se puede observar en la tabla 6.3.

Cuadro 6.3: Rendimiento del subconjunto 2

Subconjunto 2						
Modelo	F1	σ de F1	Precision	σ de Precision	Recall	σ de Recall
Random Forest	0.864636	0.048872	0.831993	0.085359	0.910655	0.074342
Support Vector Machine	0.879260	0.051013	0.843616	0.107793	0.933466	0.075063
Artificial Neural Network	0.841746	0.053205	0.828445	0.108592	0.874847	0.090239

Capítulo 7

Conclusiones

1. Se logró identificar SNPs de interés que permitieron reducir el conjunto original en subconjuntos significativos pero pequeños para la predicción del resultado del tratamiento terapéutico contra la leishmaniasis.

2. Se logró encontrar subconjuntos interesantes que dieron como resultado un buen desempeño evaluando las distintas métricas.

3. Los SNPs más representativos tienen importancia para predecir si un paciente puede curarse o no sometiéndose a este tratamiento de acuerdo a las métricas encontradas

4. La selección de parámetros juega un papel importante para encontrar los mejores modelos de aprendizaje, estableciendo una conexión efectiva entre los SNPs representativos y la respuesta al tratamiento de la leishmaniasis.

5. Los SNPs identificados en la tabla 6.1 mostraron una fuerte asociación con la predicción del desenlace terapéutico al repetirse en múltiples subconjuntos de SNPs.

6. La presencia repetida de SNPs en diferentes subconjuntos sugiere una posible implicación en la señalización celular en pacientes con leishmaniasis, lo que podría influir en su respuesta a la enfermedad y a los tratamientos. Algunos individuos podrían tener variantes genéticas que los predisponen a una respuesta terapéutica mejor o peor.

7. Los SNPs identificados constituyen un punto de partida para investigaciones futuras. Podrían llevarse a cabo estudios más detallados para comprender cómo estas variantes genéticas específicas afectan la señalización celular y la respuesta inmunitaria en pacientes con leishmaniasis.

8. El modelo Random Forest destacó como el más eficaz, maximizando el recall y reduciendo los Falsos Positivos en la predicción de la recuperación de los pacientes sometidos al tratamiento de la leishmaniasis.

9. A pesar de que el modelo Support Vector Machine no fue el mejor en términos de rendimiento, también se presenta como una opción viable debido a su desempeño cercano al del Random Forest.

7.1. Trabajo futuro

- Explorar más modelos de aprendizaje no supervisado para detectar otros posibles subconjuntos que permitan determinar el desenlace terapéutico del tratamiento de la leishmaniasis de manera efectiva.
- Explorar más modelos de aprendizaje supervisado para comparar los resultados obtenidos para dar una mayor visión del rendimiento de los subconjuntos.
- Entrenar y evaluar con otro conjunto de datos de SNPs para lograr identificar otros SNPs representativos que influyan en el desenlace terapéutico del tratamiento de la leishmaniasis.

Bibliografía

[1] Federico Oldoni and Jutta Palmen and Claudia Giambartolomei and Philip Howard and Fotios Drenos and Vincent Plagnol and Steve E. Humphries and Philippa J. Talmud and Andrew J.P. Smith (2016) Post-GWAS methodologies for localisation of functional non-coding variants: ANGPTL3. Available: https://www.sciencedirect.com/science/article/pii/S0021915015302392section-cited-by

[2] https://www.who.int/es/news-room/fact-sheets/detail/leishmaniasis

[3] T. S. Ferreira et al., "Diagnostic Classification of Cases of Canine Leishmaniasis Using Machine Learning," Sensors, vol. 22, no. 9, p. 3128, Apr. 2022, doi: 10.3390/s22093128.

[4] Malgorzata Maciukiewicz, Victoria S. Marshe, Anne-Christin Hauschild, Jane A. Foster, Susan Rotzinger, James L. Kennedy, Sidney H. Kennedy, Daniel J. Müller, Joseph Geraci, GWAS-based machine learning approach to predict duloxetine response in major depressive disorder, Journal of Psychiatric Research, Volume 99, 2018, Pages 62-68, ISSN 0022-3956, https://doi.org/10.1016/j.jpsychires.2017.12.009.

[5] Juan Enrique Sánchez Lasheras, Sergio Luis Suárez Gómez, Jesús Daniel Santos, Gemma Castaño-Vinyals, Beatriz Pérez-Gómez & Adonina Tardón (2019) A multivariate regression approach for identification of SNPs importance in prostate cancer, Journal of Experimental Theoretical Artificial Intelligence, 31:6, 817-828, DOI: 10.1080/0952813X.2018.1552319

[6] Mahalingam, Sundar, Ritika Kabra, and Shailza Singh. Çonstruction of Feed Forward MultiLayer Perceptron Model For Genetic Dataset in Leishmaniasis Using Cognitive Computing."bioRxiv (2018): 411363.

[7] J Soto. "Leishmaniasis cutánea y mucosa"FUNDERMA, 2018

[8] R. M. de Oliveira, S. de Araújo Melo, T. d. Penha-Silva, F. Almeida-Souza, and A. LuciaAbreu-Silva, .Alternative Treatment for Leishmaniasis", in Leishmaniases as Re-emerging Diseases. London, United Kingdom: IntechOpen, 2018 [Online]. Available: https://www.intechopen.com/chapters/60506 doi: 10.5772/intechopen.75895

[9] National Human Genome Research Institute https://www.genome.gov/es/genetics-glossary/Estudio-de-asociacion-de-genoma-completo

[10] Aronson N, Herwaldt BL, Libman M, et al: Diagnosis and treatment of leishmaniasis: Clinical Practice Guidelines by the Infectious Diseases Society of America (IDSA) and the American Society of Tropical Medicine and Hygiene (ASTMH). Clin Infect Dis 63 (12):e202-e264, 2016. doi: 10.1093/cid/ciw670