UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL PARANÁ

Tesis de Maestría

# GENERACIÓN DE DATOS SINTÉTICOS PARA SELECCIÓN DE CARACTERÍSTICAS CON ALGORITMOS GENÉTICOS

Por: Claudio Sebastian Castillo

Tesis presentada en cumplimiento parcial de los requerimientos para la obtención del grado académico de: MAGISTER EN MINERÍA DE DATOS

Director: Dr.Matías Gerard
Co-Director: Dr.Leandro Vignolo

Paraná, Argentina
Mayo, 2025

A mi corazón de 4/4; Morella, Sofía, Joquín y Manuel. A mi musa Verónica.

# Resumen

Generación de Datos Sintéticos para Selección de Características con Algoritmos Genéticos

Claudio Sebastian Castillo

En problemas de aprendizaje automático, la disponibilidad de datos muestrales influye significativamente en los procesos de selección de características, volviéndose especialmente crítica en escenarios de alta dimensionalidad y presencia de ruido. En general, cualquier método de selección de características es sensible a la cantidad de datos disponibles. Cuando la selección se realiza mediante Algoritmos Genéticos (AG), la falta de datos impacta negativamente en la función de aptitud, limitando así la eficacia del algoritmo. Para abordar este problema, proponemos integrar el proceso de selección de características con un paso previo de generación sintética de datos mediante Autocodificadores Variacionales (AV). Este enfoque permite aumentar de forma artificial la información disponible. El flujo de procesamiento resultante combina AV y AG, y los aplica a procesos de selección exigentes. De esta forma, se obtienen mejores resultados en comparación con el procesamiento sin aumentación. En este estudio se detalla la integración de AV y AG, y se describen los experimentos realizados sobre cinco conjuntos de datos ampliamente utilizados en la literatura: Leukemia, Gisette, Mandelon, GCM y ALL Leukemia. Asimismo, se presentan las arquitecturas de los modelos empleados en la integración.

# Abstract

Synthetic Data Generation for Feature Selection with Genetic Algorithms

Claudio Sebastian Castillo

In machine-learning tasks, the amount of available training data strongly affects feature-selection procedures and becomes particularly critical in high-dimensional, noisy settings. Most feature-selection methods are sensitive to sample size; when Genetic Algorithms (GA) drive the search, data scarcity degrades the fitness function and, consequently, the algorithm's effectiveness. To mitigate this problem, we integrate feature selection with a preceding synthetic-data generation step based on Variational Autoencoders (VAE). The resulting pipeline —VAE augmentation followed by GA-based selection— artificially expands the information available and applies it to demanding feature-selection scenarios, yielding superior results compared with the same process without augmentation. We detail the VAE–GA integration and report experiments on five benchmark datasets widely cited in the literature: Leukemia, Gisette, Madelon, GCM, and ALL Leukemia. We also describe the model architectures used in each stage of the pipeline.

# Introducción

En el campo del aprendizaje automático, la selección de características es una tarea crítica que puede determinar el éxito o fracaso de un modelo predictivo. La alta dimensionalidad y la complejidad inherente a los conjuntos de datos reales hacen que la selección de un subconjunto óptimo de características sea, muchas veces, un paso ineludible para el aprendizaje efectivo.

En este contexto, los Algoritmos Genéticos (AG) se han consolidado como una herramienta poderosa para resolver problemas de optimización complejos, incluida la selección de características (Vignolo y Gerard 2017). Estos algoritmos, inspirados en la evolución natural, son capaces de explorar grandes espacios de búsqueda de manera efectiva, proporcionando soluciones suficientemente buenas en una amplia variedad de escenarios. Por ello, los AG han sido ampliamente utilizados en problemas de selección, demostrando su eficacia en la identificación de subconjuntos relevantes de características en datos de alta dimensionalidad.

Sin embargo, la eficacia de los AG depende de la disponibilidad de suficientes datos para evaluar las soluciones en competencia. En contextos donde los datos son limitados, donde cualquier método de selección de características es sensible a esta limitación, también los AG pueden verse afectados en su capacidad para seleccionar características relevantes, produciendo soluciones subóptimas o inestables. Esta limitación es especialmente crítica en problemas de alta dimensionalidad y bajo número de muestras, donde la función objetivo que guía la búsqueda de soluciones puede degradarse significativamente (Bolón-Canedo, Sánchez-Maroño, y Alonso-Betanzos 2015).

Por esta razón, la investigación de estrategias que mitiguen las limitaciones impuestas por la escasez de datos se ha convertido en un área de interés creciente en el subcampo de la selección de características. Una de las técnicas emergentes en este ámbito es la aumentación de datos mediante Autocodificadores Variacionales (AV). Los AV, como modelos generativos, tienen la capacidad de crear muestras sintéticas que mantienen las propiedades fundamentales de los datos originales, conviertiéndolos en una herramienta prometedora para mejorar la capacidad de los AG en la selección de características.

El problema central de la tesis que aquí presentamos gira, precísamente, en la restricciones que la escacez de datos impone a la función objetivo que guía la búsqueda de soluciones de los AG, y cómo superarlas usando AV. Así, la pregunta central del trabajo es: ¿cómo puede la aumentación de datos mediante autocodificadores variacionales mejorar el desempeño de los algoritmos genéticos en la selección de características?

Cabe destacar que este desafío y su eventual solución son importantes por varias razones. La selección de características no solo condiciona, como ya se mencionó, la precisión de los modelos predictivos, también afecta la eficiencia computacional y la interpretabilidad de los resultados. En problemas de alta dimensionalidad, la posibilidad de reducir el número de características relevantes sin perder información útil puede marcar la diferencia entre un modelo efectivo y uno ineficaz, entre uno interpretable y uno de caja negra. Por lo tanto, mejorar este proceso mediante la integración de técnicas de aumentación de datos puede tener un impacto significativo en diversas aplicaciones prácticas, desde la biología molecular hasta la ingeniería y las ciencias sociales (El-Hasnony et al. 2020).

La hipótesis que hemos llevado a prueba ha sido que la aumentación de datos mediante AV mejora la capacidad de selección de características de los AG, favoreciendo

la evaluación de soluciones en competencia y permitiendo la identificación de subconjuntos de características más relevantes y estables en contextos de escasez muestral. Para evaluarla, hemos propuesto un trabajo experimental que exploró la integración de estas dos técnicas en un marco unificado. La idea de combinar la generación de datos sintéticos mediante AV con la selección de características mediante AG, estaba orientada a buscar combinaciones sinérgicas y eficaces entre modelos que mejorasen la selección de características. A estos fines, trabajamos con cinco conjuntos de datos de referencia, representativos de distintos contextos y niveles de complejidad, para evaluar el desempeño de los modelos propuestos.

A lo largo de este documento, describiremos el proceso de investigación llevado a cabo, desde los estudios iniciales hasta los experimentos finales, pasando por el diseño y construcción de un modelo genérico de AV, y su adaptación a los datasets elegidos y la creación de una estructura combinada de AV + AG para la selección de características. Los resultados obtenidos en cada etapa se analizarán y discutirán en detalle, con el objetivo de identificar las ventajas y limitaciones de la propuesta, así como posibles áreas de mejora y futuros trabajos.

Al finalizar el documento, esperamos poder justificar la eficacia de la aumentación de datos mediante AV en la selección de características, demostrando que esta técnica puede mejorar significativamente el desempeño en contextos de escasez. Además, esperamos identificar las condiciones y contextos en los que esta técnica es más efectiva.

El documento está estructurado de la siguiente manera: en el Capítulo 1 se presenta el problema de la selección de características en contextos de alta dimensionalidad y escacez muestral. En el Capítulo 2 se hace una revisión de modelos clásicos aplicados a los datos que forman parte de nuestra investigación, su evaluación y resultados. En el Capítulo 3 se presentan los modelos AV, sus bases teóricas y los experimentos realizados para construir la arquitectura final empleada en nuestra investigación. En el Capítulo 4 se aborda la integración de los modelos AV y AG en una estructura combinada, su configuración y los resultados experimentales obtenidos de su aplicación. Finalmente, en el Capítulo 5, se presentan las conclusiones de la investigación, así como posibles líneas futuras de trabajo.

Por último, en tiempos de grandes debates sobre los riesgos de los modelos generativos, esperamos poder brindar a nuestros lectores un ejemplo funcional de su estructura, iluminar ciertos principios y evidenciar sus límites. Si nuestro trabajo permite resaltar la sobriedad de sus mecanismos internos, habremos cumplido el anhelo de hacer menos opaca su belleza y más clara sus posibilidades.

# Índice

Capítulo 1

# El problema de la selección de características

En el presente capítulo abordamos el problema de la selección de características, su importancia y desafíos. En ese marco, repasamos ciertas dificultades asociadas a los datos, como por ejemplo la alta dimensionalidad y el desbalance de clases, y veremos cómo ellas impactan en la selección de características. Luego describimos brevemente distintos enfoques para la selección de características, como así también ventajas y desventajas de cada uno. Finalmente, vinculamos la selección de características con el aporte fundamental de nuestro trabajo asociado a la generación sintética de datos.

## 1.1. Estrés, ignorancia y selección de características

Un punto de partida difícil de controvertir en el mundo actual del aprendizaje automático es que la cantidad de información disponible para investigar ha crecido dramáticamente en los últimos veinte años (Li and Zhang 2023). Conforme la vida se digitaliza, y la información almacenada en los sistemas aumenta, la generación de conocimiento parece menos determinada por el viejo problema de la escasez de factores y más por su nueva situación de abundancia.

En este contexto, el desafío de trabajar con datos de alta dimensionalidad ha motivado la aparición de una serie de técnicas dirigidas a generar conocimiento y resultados precisos en escenarios complejos debido a la abundancia de información. Así, bajo el nombre de selección de características nació un area de estudio que busca resolver, precisamente, el problema de discernir sistemáticamente la información relevante de aquella que no lo es cuando se trabaja con muchas variables.

Según Bolón-Canedo et al., el origen de este campo se remonta a los años '60, cuando Hughes, usando un modelo paramétrico, estudió la precisión de un clasificador bayesiano en función del número de características utilizado para predecir una variable objetivo (Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015). Por nuestra parte, entendemos que incluso se puede ir más lejos, a los años '30, cuando Fisher reprochaba el estrés con el que se buscaba conocimiento manipulando variables aisladas en lugar de mirar sus interacciones, proponiendo una investigación experimental más abarcativa y eficiente que permitiera obtener mayor conocimiento con la misma cantidad de observaciones (Fisher 1935).

Más allá de su origen, el poderoso impulso de considerar espacios de búsqueda cada vez más amplios y problemas cada vez más complejos, ha contribuido a que la selección de características se convierta en un campo de estudio activo e importante.

En efecto, hoy no es extraño encontrar investigaciones donde los objetos de estudio superen las decenas de miles de dimensiones, como sucede en la investigación biomédica. Allí, los microarrays de ADN son ejemplos representativos de datos de alta dimensionalidad, que constituyen fuentes vitales de información en problemas que involucran expresión génica (Almugren and Alshamlan 2019). En sentido similar, datos de video -presentes en múltiples aplicaciones-, datos financieros, información vinculada a interacciones sociales, entre otros, son ejemplos del tipo de problema que justifican el uso de técnicas de selección de características (El-Hasnony et al. 2020).

Podemos definir a la selección de características como el proceso de detectar las características relevantes y descartar aquellas irrelevantes o redundantes, con el objetivo de obtener un subconjunto que describa adecuadamente un problema dado con una degradación mínima del rendimiento (Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015). Aquí rendimiento se refiere a la medida de evaluación empleada para juzgar los resultados del proceso de selección. Los distintos enfoques para selección de características adhieren a la premisa de que podemos separar la información relevante de aquella que no lo es para predecir una variable objetivo, y por ende mejorar el desempeño de los modelos de aprendizaje.

Nótese aquí que el proceso de selección de características es un proceso sistemático. Gran parte de su esfuerzo se centra en disponer de un método para discriminar, de manera precisa y controlada, la información relevante de la irrelevante o redundante. La precisión y el control destacan como dos características fundamentales debido a que la selección de características tiene lugar en un escenarios donde se ignora cual es el aporte de cada variable a la resolución del problema. Por eso, resulta necesario contar con un criterio de evaluación claramente definido que permita examinar de forma objetiva y cuantificable el valor real de cada variable y sus interacciones.

Bajo esa perspectiva, la selección de características busca determinar un subconjunto de atributos que satisfaga uno de los siguientes criterios (Vignolo and Gerard 2017):

1. El subconjunto con un tamaño especificado que maximice la precisión de la predicción.
2. El subconjunto de menor tamaño que satisfaga un requisito de precisión mínima.
3. El subconjunto que logre el mejor compromiso entre dimensionalidad y precisión.

El criterio a elegir dependerá de los objetivos del estudio y de las características del problema. Mas allá de eso, es fácil advertir que la selección de características supone como ventaja la reducción del espacio de los datos, y es un remedio a la alta dimensionalidad.

En efecto, en el contexto del aprendizaje automático es común enfrentar problemas representados por grandes conjuntos de variables, vicisitud que se asocia con la maldición de la dimensionalidad (Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015). En general, este fenómeno se presenta por la alta demanda computacional y costos asociados con la optimización de dichos espacios, volviendo a ciertos problemas intratables desde el punto de vista práctico. Para abordarlo, una alternativa posible es la reducción dimensional, que consiste en realizar una transformación del sistema de coordenadas (lineal o no lineal) para representar los datos mediante un conjunto reducido de características, construidas como combinaciones de las variables originales. El resultado de esta transformación, es una matriz de menor dimensión, que puede usarse de manera más eficiente para modelar el problema objetivo, facilitando su interpretación y comunicación.

Dicho lo anterior, es preciso reconocer que la abundancia de información trae consigo una serie de desafíos que impactan en la selección de características. En el siguiente apartado describiremos aquellos más relevantes para nuestro estudio.

## 1.2. Las múltiples caras de los datos problemáticos

### 1.2.1. Alta dimensionalidad y escasez muestral

En primer lugar, un problema frecuente en el aprendizaje automático se presenta cuando disponemos de pocas muestras en un espacio vectorial de alta dimensionalidad. Es decir, cuando el número de muestras ($m$) es menor que el número de características ($n$), siendo $n \gg m$. Como vimos, esta situación es común en el campo del análisis genético, el procesamiento de información médica, procesamiento de video, entre otros. Como veremos mas adelante, tres de los datasets elegidos para nuestro estudio -vinculados al ámbito biomédico- se encuentran en esta situación.

La escasez de muestras dificulta el tratamiento de la información, ya que la cantidad de datos disponibles para entrenar un modelo de clasificación o regresión es insuficiente para capturar la variabilidad inherente a los datos. En tales circunstancias, el modelo resultante tiende a sufrir severas limitaciones, bien sea sobreajustándose a las observaciones disponibles (llevando a modelos poco generalizables), o bien resultando incapaz de capturar la estructura subyacente de los datos (lo que puede llevar a modelos poco precisos).

Por ejemplo, en el campo de la clasificación de perfiles de expresión génica, el problema se considera difícil de resolver debido a la complejidad que supone identificar los genes que contribuyen a la aparición de ciertas condiciones - como por ejemplo: cáncer, diabetes, entre otros (Almugren and Alshamlan 2019). Dada la gran cantidad de genes presentes en estos supuestos (muchos de ellos irrelevantes), entrenar un modelo con todos ellos puede conducir a resultados erróneos. Este desafío, comúnmente se ve acentuado por la baja cantidad de observaciones disponibles y por el hecho de que los genes se encuentran altamente correlacionados. Dichas características dificultan el adecuado funcionamiento de los métodos clásicos de aprendizaje automático, cuyo rendimiento se asocia a la cantidad de muestras disponibles.

Ante estos problemas, la selección de características tiene severas limitaciones, ya que la alta dimensionalidad y bajo número de muestras dificultan la distinción entre información relevante e irrelevante.

### 1.2.2. Desbalance de clases

El desbalance de clases constituye otro problema importante y frecuente en los desafíos actuales que se presentan para el aprendizaje automático. El mismo sucede cuando la distribución de las clases en el conjunto de datos es altamente desigual: típicamente la cantidad de observaciones de una clase mayoritaria supera ampliamente a las de una o más clases minoritarias. Este problema puede darse en combinación con alta dimensionalidad y escacez muestral, agravando la dificultad de cualquier tarea de aprendizaje maquinal, incluida la selección de características.

El desbalance de clases es un problema relevante en aplicaciones donde la o las clases minoritarias son precisamente las de mayor interés, como por ejemplo: el diagnóstico de enfermedades raras, la detección de fraudes bancarios, la identificación de intrusiones en redes y la predicción de fallos en equipos técnicos. En supuestos como estos,

los clasificadores que no contemplen un tratamiento explícito del problema tienden a sesgarse hacia la clase mayoritaria, pudiendo alcanzar una alta precisión global mientras fallan en la detección de los casos más importantes pero menos frecuentes.

Para atacar este problema se han propuesto diversas soluciones que pueden categorizarse -como señala Bolón-Canedo et al.- en tres grupos:

1. Muestreo de datos: Este enfoque modifica las muestras de entrenamiento para producir una distribución de clases más balanceada. Las técnicas tradicionales incluyen: submuestreo (undersampling) donde se crea un subconjunto del conjunto original eliminando instancias; sobremuestreo (oversampling) donde se genera un superconjunto replicando instancias existentes o creando nuevas, y finalmente, métodos híbridos que combinan ambas técnicas de muestreo.

2. Modificación algorítmica: Este enfoque adapta los métodos de aprendizaje para que sean más sensibles a los problemas de desbalance. Por ejemplo, el uso de la distancia de Hellinger como criterio de división en árboles de decisión, que ha demostrado ser insensible al sesgo, capturando la divergencia en las distribuciones sin ser dominada por las probabilidades previas de clase.

3. Aprendizaje sensible al costo: Esta solución puede incorporar elementos a nivel de datos, a nivel de algoritmos, o ambos a la vez, asignando costos más altos a la clasificación errónea de ejemplos de la clase positiva (minoritaria). Así, en el diagnóstico médico, resulta más importante reconocer la presencia de una enfermedad rara que su ausencia, por ello el costo de un falso negativo es mayor que el de un falso positivo.

El desbalance de clases también supone un desafío para la selección de características. Ello es así, en la medida que dichos métodos suponen la búsqueda de aquellas dimensiones que maximizan la separación de clases en su conjunto, sin ponderar sus diferencias para capturar información de cada clase en particular. Esto ocurre porque, en escenarios con distribución fuertemente desequilibrada, las técnicas de evaluación tienden a priorizar la identificación de la clase mayoritaria, reduciendo la relevancia de las características que serían valiosas para discriminar a las minoritarias. Como resultado, se descarta información relevante para la predicción de los casos poco frecuentes, y se seleccionan características que no son representativas para todas las clases por igual.

Junto a los problemas mencionados sobre alta dimensionalidad y desbalance de clases, la complejidad y el ruido de los datos también representan desafíos importantes en el tratamiento de la información que debemos considerar en el presente estudio.

### 1.2.3. Complejidad

Respecto a la complejidad, se refiere a la dificultad de identificar las fronteras de decisión entre clases, que pueden manifestarse en tres aspectos fundamentales (todos ellos pueden coexistir en un mismo problema):

1. Ambigüedad de clases: Surge cuando los casos no pueden distinguirse utilizando las características disponibles, ya sea porque los conceptos de clase están mal definidos y por lo tanto son intrínsecamente indistinguibles, o porque las características seleccionadas son insuficientes para discriminar las clases.

2. Complejidad de fronteras: Se refiere a situaciones donde los límites entre clases requieren una descripción extensa de la frontera de decisión, llegando a demandar, en el caso extremos de complejidad, la enumeración exhaustiva de todos los puntos con sus etiquetas de clase. Este aspecto de dificultad se debe a la naturaleza del problema y no a la muestra o selección de características, indicando que una completa separación entre clases es un problema difícil de resolver.

3. Dispersión de muestras: La combinación de pocas muestras y alta dimensionalidad genera una dispersión que dificulta la generación de fronteras de decisión.

Aunque los casos 1 y 2 son problemas ante los cuales la selección de características no puede ofrecer una solución efectiva, el caso 3, asociado con alta dimensionalidad y escasez muestral, podría ser mitigado por estrategias de aumentación de datos.

### 1.2.4. Ruido

Los datos del mundo real siempre están expuestos a imperfecciones, ruido, proveniente de entornos dinámicos donde las dimensiones de interés de un fenómeno coexisten en espacios de interacciones permanentes. Así, aún considerando un escenario artificial, completamente libre de interferencia, la imperfección puede provenir de diversas fuentes como dispositivos de medición defectuosos o limitados, errores de transcripción o irregularidades en la transmisión de información.

Ante tales circunstancias, existen cuatro enfoques principales para abordar estas imperfecciones en los conjuntos de datos:

1. Conservación del ruido: En este enfoque, se mantiene el conjunto de datos tal como está, con sus instancias ruidosas. Los algoritmos que utilizan los datos se diseñan para ser robustos, es decir, capaces de tolerar cierta cantidad de ruido. Una estrategia común es desarrollar algoritmos que eviten el sobreajuste del modelo (como sucede en el caso de los árboles de decisión) mediante técnicas de poda.

2. Eliminación de datos ruidosos: Este método implica descartar las instancias que se consideran ruidosas según ciertos criterios de evaluación. El clasificador se construye utilizando únicamente las instancias retenidas en un conjunto de datos más pequeño pero más limpio. Sin embargo, este enfoque presenta dos debilidades significativas:

   - Al eliminar instancias completas, se puede descartar información potencialmente útil, como valores de características no corrompidos.
   - Cuando existe una gran cantidad de ruido, la información restante en el conjunto de datos limpio puede resultar insuficiente para construir un clasificador eficaz.

3. Transformación de datos: Este enfoque busca corregir las instancias ruidosas en lugar de eliminarlas. Las instancias identificadas como ruidosas se reparan reemplazando los valores corrompidos por otros más apropiados, y luego se reintroducen en el conjunto de datos.

4. Reducción de datos: Esta estrategia implica reducir la cantidad de datos mediante la agregación de valores o la eliminación y agrupación de atributos redundantes. La reducción de dimensionalidad es una de las técnicas más populares para eliminar características ruidosas (es decir, irrelevantes) y redundantes.

Vistos los problemas que enfrentamos con los datos, pasemos a describir, brevemente, los distintos enfoques que se han propuesto para sortearlos empleando estrategias de selección de características.

## 1.3. Enfoques para la selección de características

Podemos agrupar los métodos de selección de características en tres grandes grupos, representados en el diagrama de la Figura 1.1.:

- Filtros,
- Wrappers o métodos envolventes, y
- Embedded o métodos embebidos.

Los métodos de Filtro se basan en las características generales de los datos de entrenamiento y realizan la selección de características como un paso de preprocesamiento independiente del algoritmo de aprendizaje. Este enfoque es ventajoso por su bajo costo computacional, rapidez y escalabilidad, pero deja sin resolver el problema apuntado por Fisher sobre la interacción entre variables. Los métodos de filtro se pueden sub-clasificar en univariados y multivariados (Solorio-Fernández, Carrasco-Ochoa, and Martínez-Trinidad 2020). Los primeros analizan cada característica de manera independiente con el fin de obtener una lista ordenada. Este tipo de métodos puede identificar y eliminar eficazmente características irrelevantes, pero no son capaces de eliminar las redundantes, ya que no consideran posibles dependencias entre las características. Ejemplos de estos métodos son: la evaluación utilizando la distribución chi-cuadrado, la ganancia de información, entre muchos otros (Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015). Los métodos filtro multivariados evalúan la relevancia de las características de forma conjunta en lugar de hacerlo individualmente. Por ejemplo, el método de selección hacia adelante (forward selection) y hacia atrás (backward selection) son métodos de filtro multivariados que sucesivamente agregan y eliminan características para obtener un subconjunto óptimo. Los métodos multivariados pueden manejar características redundantes e irrelevantes; por lo tanto, en muchos casos, la precisión alcanzada por los modelos empleando subconjuntos seleccionados con estos métodos puede ser mayor. Otro método filtro multivariado puden ser el método basado en el análisis de correlación (Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015).

Los métodos Wrappers o Envolventes involucran un algoritmo de aprendizaje como caja negra y consisten en usar su resultado para evaluar la utilidad relativa de subconjuntos de características. Aquí, el algoritmo de selección utiliza el método de aprendizaje como una subrutina, para encontrar los subconjuntos de características más relevantes. Esta estrategia es capaz de reconocer variables relevantes y capturar dependencias entre ellas, pero a un costo computacional mayor que los otros métodos. En ese sentido enfrenta desafíos importante cuando el conjunto de datos es de alta dimensionalidad, ya que evaluar $2^n$ subconjuntos de características puede resultar en un problema intratable. Por esa razón, se recurre comunmente a heurísticas de búsqueda capaces de encontrar soluciones adecuadas sin explorar todo el espacio del problema (R. Zhang et al. 2019).

Los métodos Embedded o Embebidos realizan la selección de características durante el proceso de entrenamiento de un modelo de aprendizaje, y suelen ser específicos para determinados algoritmos (e.g. árboles de decisión y métodos derivados de ellos, eliminación recursiva de características mediante SVM, entre otros). A diferencia de

los métodos de filtro y wrappers, los métodos embebidos no separan la selección de características del proceso de entrenamiento, sino que la realizan de manera simultánea. En este caso, optimizan una función de pérdida regularizada con respecto a dos conjuntos de parámetros: los parámetros del algoritmo de aprendizaje y los parámetros que indican las características seleccionadas.(Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015). Este enfoque es capaz de capturar dependencias a un costo computacional menor que los wrappers.

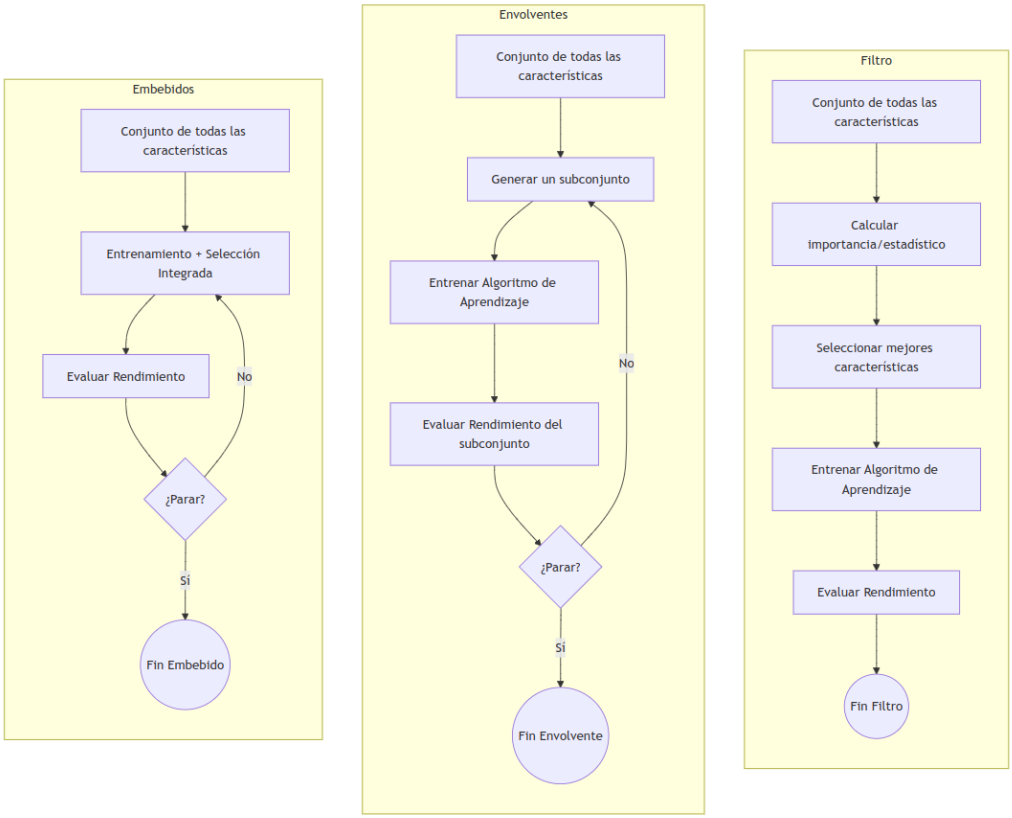


Figura 1.1 Diagrama de enfoques para la selección de características

Bolón-Canedo et al. (Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015) nos invitan a pensar que no existe un método que sea superior a los otros de manera general, sino que cada uno tiene sus ventajas y desventajas, y se ajustan distinto a diferentes tipos de contexto. De los tres enfoques, los métodos de filtro son los únicos que son independientes del algoritmo de aprendizaje, lo que les permite ser rápidos y eficientes computacionalmente. Sin embargo, los métodos de filtro no consideran la correlación entre características, lo que puede llevar a la selección de características irrelevantes o redundantes.

Los métodos de envolventes y embebidos, por su parte, consideran la correlación entre características y también entre características y etiquetas de clase, lo que les permite identificar patrones relevantes durante la fase de aprendizaje. Sin embargo, estos métodos son más complejos computacionalmente y requieren evaluación iterativa del subconjunto seleccionado de características.

En el caso de los métodos envolventes, a medida que el número de características aumenta, el costo computacional se incrementa exponencialmente, lo que limita su aplicación en problemas de alta dimensionalidad (Bolón-Canedo, Sánchez-Maroño, and Alonso-Betanzos 2015). En línea con esta dificultad, se han propuesto distintas

formas de superar el problema, como el uso de heurísticas de búsqueda para evitar exploraciones exhaustivas. Entre ellas, se han propuesto a los algoritmos genéticos (AG) como herramientas eficaces para la búsqueda de soluciones óptimas (Vignolo and Gerard 2017). El propósito de este enfoque es explorar el espacio de soluciones de manera eficiente, permitiendo la identificación de subconjuntos de características que sean óptimos en términos de rendimiento.

## 1.4. Selección, algoritmos genéticos y datos sintéticos

Hasta aquí hemos visto que problemas tales como la dimensionalidad, el desbalance de clases y el ruido plantean verdaderos desafíos para los procesos de aprendizaje automático, y de qué manera la selección de características puede ser una herramienta para mitigarlos. Asimismo, hemos descrito los distintos enfoques para la selección de características, resaltando el hecho de que los métodos envolventes utilizan heurísticas como los AG para realizar búsquedas eficientes en espacios de gran dimensionalidad. Aquí surgió el problema de la escasez de datos y su impacto negativo sobre la evaluación de los subconjuntos de características; una importante limitación para los métodos envolventes de selección. En este punto final del Capítulo 1, veremos de qué manera la generación sintética de datos puede resolver el problema de la escasez muestral, y de esa forma mejorar los procesos de selección de características basados en métodos envolventes. Precisamente, este eje recorre el aporte central que realizaremos en la presente tesis.

Como dijimos, en el contexto de los métodos envolventes, los AG representan una solución al problema de la evaluación iterativa de subconjuntos de características en datos de alta dimensionalidad. Tal evaluación debe replicarse por cada subconjunto a lo largo del espacio de búsqueda, lo que resulta en un costo computacional elevado cuando el conjunto de variables a evaluar es grande (R. Zhang et al. 2019).

Inspirados en los principios de la evolución natural, los AG son capaces de realizar esa exploración eficientemente, iterando sobre múltiples generaciones de individuos. Exploraremos esto en detalle en el Capítulo 4, sin embargo, cabe señalar desde ahora que dicha capacidad está determinada en gran medida por el conjunto de operadores evolutivos con los que trabajan los AG: selección, cruce y mutación. La selección, paso crítico del proceso de búsqueda, otorga mayor probabilidad de reproducción a los individuos mejor evaluados. En este caso, los subconjuntos de características con mayor capacidad de discriminación son seleccionados en virtud de su desempeño (evaluado por una función objetivo representada en este contexto por un modelo de aprendizaje automático). El cruce, que combina partes de la solución de dos individuos para generar descendientes potencialmente mejores. Y la mutación, que introduce variaciones aleatorias y evita la convergencia prematura hacia subóptimos locales. A través de iteraciones sucesivas, estos operadores permiten explorar de manera equilibrada tanto regiones prometedoras del espacio de búsqueda como soluciones novedosas, optimizando la búsqueda de subconjuntos de características incluso en escenarios con alta dimensionalidad y complejidad.

Dicho lo anterior, es preciso advertir que la eficacia de los AG depende en gran medida de la disponibilidad de datos suficientes para evaluar el valor de los individuos a lo largo de las generaciones. En efecto, la falta de datos para estimar de forma confiable ese desempeño, agrega incertidumbre al momento de la selección, que es el paso crítico del proceso de búsqueda. Esta estimación es realizada por un modelo de aprendizaje automático con el que se evalúa el desempeño (fitness) de cada individuo (por ejemplo,

un clasificador basado en árboles de decisión, máquinas de soporte vectorial, o un perceptrón multicapa). Por eso, con respecto a este componente en particular, el AG se encuentra en la misma situación que cualquier otro método de optimización, en el sentido de que la escasez de datos afecta directamente la calidad de las predicciones utilizadas para guiar la búsqueda. Cuando los datos son insuficientes o no representan adecuadamente la variabilidad del problema, aumenta la incertidumbre en la fase de evaluación de los individuos, comprometiendo la confiabilidad del proceso de selección y la eficacia general del AG.

Es en este escenario, ante tan importante restricción, es donde entra nuestra contribución. Entendemos que la generación sintética de datos puede ser una herramienta eficaz para resolver el problema de la escasez muestral, desbalance de clases y ruido, y por ende, mejorar la capacidad de los AG para explorar y discriminar buenas soluciones en espacios de gran dimensionalidad. A lo largo de la presente tesis, mostraremos cómo dicha estrategia puede ser implementada en el contexto de la selección de características basada en AG.

Ahora bien, existen varias técnicas de aumentación de datos, desde las más tradicionales, como el sobremuestreo (SMOTE (Blagus and Lusa 2013)), hasta las más modernas, basadas en modelos generativos (por ejemplo, usando redes GANS (Fajardo et al. 2021)). Entre tales opciones los autocodificadores variacionales (AV) (Kingma and Welling 2019) han adquirido popularidad gracias a su buen desempeño, incluso superando a otros métodos de aumentación. Estos modelos, con arquitectura encoder-decoder, han demostrado ser especialmente adecuados para escenarios donde se busca preservar la estructura estadística subyacente de los datos y, a la vez, generar muestras lo suficientemente diversas para captar la variabilidad del problema. Abordaremos esto en detalle en el Capítulo 3, sin embargo, cabe adelantar aquí que estos modelos se basan en la estimación y reconstrucción de distribuciones latentes continuas, lo que les permite reconocer patrones relevantes y producir muestras sintéticas similares a los datos originales. Por estas propiedades deseables que presenta su funcionamiento, los AV se han usado para aumentación de datos en el campo del tratamiento de imágenes (Fajardo et al. 2021; Ai et al. 2023; Khmaissia and Frigui 2023; Kwarciak and Wodzinski 2023), texto (Y. Zhang et al. 2019) , habla (Blaauw and Bonada 2016; Latif et al. 2020) y música (Roberts et al. 2019), y distintos formatos de datos: tabulares (Leelarathna et al. 2023), longitudinales (Ramchandran et al. 2022) y grafos (Liu et al. 2018).

La versatilidad y capacidad de los AV, los convierte en una herramienta potencialmente transformadora para los procesos de selección de características basada en AG, abriendo la posibilidad de aplicaciones en escenarios muy diversos, y problemas que involucran distintas modalidades de datos.

Volviendo a los problemas señalados, la generación de muestras sintéticas puede realizar un importante aporte al tratamiento del desbalance de clases. En escenarios críticos con problemas multiclase y distribuciones asimétricas, la generación sintética de datos ofrece la posibilidad de reequilibrar las proporciones de cada categoría. En efecto, creando suficientes muestras representativas de las clases minoritarias, la generación sintética de datos puede mejorar la discriminación entre categorías poco representadas. Con esta estrategia, se reduciría el sesgo introducido por la disparidad de muestras y, por ende, se mejoraría el proceso de selección de características.

Por último, es importante destacar que la generación sintética de datos mediante AV también puede contribuir a mitigar el efecto del ruido en los modelos de aprendizaje.

Durante el proceso de codificación-decodificación implementado por estos modelos, el proceso de reconstrucción identifica y proyecta las características esenciales de los datos desde su dimensión original a un vector de menor dimensión, favoreciendo la reducción de información irrelevante (Kingma and Welling 2019). Este mecanismo, analizado en mayor detalle en el Capítulo 3, abonaría la importancia de los AV en la creación de muestras sintéticas no solo para reducir el impacto negativo del ruido sobre el proceso de entrenamiento, sino también para mejorar el desempeño de los métodos evolutivos en la búsqueda de características óptimas.

Todas estas razones respaldan la importancia de la generación sintética de datos para la selección de características con AG. En la presente tesis se propone una estrategia integral que vincula la selección de características y la generación sintética de datos, enfocándose en solucionar problemas habituales en aprendizaje automático, tales como la alta dimensionalidad, la escasez de muestras, la complejidad y el ruido.

Siguiendo esta línea, los próximos capítulos de la tesis se organizan de la siguiente manera:

- En el Capítulo 2, se realiza un análisis preliminar de los datos con los que trabajaremos, se describen los modelos de aprendizaje automático evaluados para servir de función objetivo en el AG que desarrollaremos, y se presentan las métricas de evaluación aplicadas a lo largo del estudio. Además, se discuten potenciales dificultades asociadas a la naturaleza de los datos.
- En el Capítulo 3, se profundiza en la teoría de los autocodificadores variacionales y en la motivación para usarlos como técnica de aumentación de datos. Se detallan los experimentos de generación sintética, los resultados obtenidos y la forma en que estos modelos capturan la estructura subyacente de la distribución original, contribuyendo así a otorgar calidad a los datos generados.
- En el Capítulo 4, se presenta el AG desarrollado, describiéndose la integración de los AV como etapa de preprocesamiento para enriquecer la selección de características. Asimismo, se exponen los experimentos realizados tanto con datos originales como con datos aumentados, demostrando el impacto de la generación sintética en el rendimiento de la selección de características.
- Finalmente, en el Capítulo 5, se exponen las conclusiones de la investigación, resaltando la relevancia de la generación sintética de datos para la selección de características en escenarios complejos, y se ofrecen perspectivas de trabajo futuro donde las técnicas propuestas podrían ampliarse o refinarse.

De este modo, se establece un marco metodológico completo que explora, integra y evalúa la sinergia entre AV, AG y selección de características, sirviendo como aporte innovador para atender las dificultades impuestas por la dimensionalidad, la escasez de datos, el desbalance de clases y el ruido en el ámbito del aprendizaje automático.

Capítulo 2

# Modelos clásicos aplicados al espacio completo de características

En este capítulo revisamos el desempeño de algoritmos clásicos en la solución de los problemas elegidos para nuestra investigación, utilizando el espacio completo de características. El objetivo de esta exploración es doble: por un lado contar con métricas de base para comparar el desempeño de nuestras soluciones, y por el otro, estudiar las características de los datasets elegidos para nuestro estudio, procurando identificar aquellos rasgos que puedan influir en el desempeño de los modelos. Para la selección de los dataset a utilizar, nos centramos en la alta dimensionalidad y la escasez muestral, el desbalance de clases y el ruido.

## 2.1. Datos elegidos en nuestro estudio

El conjunto de datos elegidos en este trabajo incluye cinco datasets: Madelon, Gisette, Leukemia, GCM y ALL Leukemia. El último de ellos, ALL Leukemia, es un dataset introducido en este trabajo iniciada la etapa de experimentación de integración entre AG y AV. Por esa razón, no está incluido en la revisión de desempeño de los algoritmos clásicos que presentaremos en este capítulo. No obstante ello, se incluye en el detalle de los datasets para que el lector tenga una idea de la variedad de problemas que se tuvieron en cuenta para validar los resultados de nuestra propuesta.

Como veremos a continuación, cada dataset plantea desafíos distintos en términos de aprendizaje, y posee distintos niveles de complejidad en su composición. El dataset Madelon es un conjunto artificial de datos con 2000 observaciones y 500 características (2000x500), donde el objetivo es resolver un problema XOR multidimensional con 5 características relevantes y 15 características corresponden a combinaciones lineales de aquellas (i.e. 15 características redundantes). Las otras 480 características fueron generadas aleatoriamente (no tienen poder predictivo). Madelon es un problema de clasificación de dos clases con variables de entrada binarias dispersas. Las dos clases están equilibradas, y los datos se dividen en conjuntos de entrenamiento y prueba. Fue creado para el desafío de Selección de Características NIPS2003[1], y está disponible en el Repositorio UCI[2]. Los datos están divididos en un conjunto de entrenamiento y un conjunto de testeo.

Como es fácil de advertir, este es un problema donde la información relevante está presente junto a información redundante y otra sin valor predictivo. Es decir, existe

[1] http://clopinet.com/isabelle/Projects/NIPS2003/.
[2] https://archive.ics.uci.edu/dataset/171/madelon.

un alto nivel de ruido en los datos, planteando importantes desafíos para los algoritmos de aprendizaje, y uno particularmente interesante para el problema de selección de características. Respecto de las dimensiones del problema, no se estaría en una situación crítica de alta dimensionalidad y escasez muestral, ya que el número de observaciones es mayor que el de características. Sin perjuicio de ello, aún queda por determinar si la cantidad de patrones disponibles es suficiente para que los algoritmos de aprendizaje puedan encontrar una solución en un contexto tan ruidoso.

Gisette es un dataset creado para trabajar en el problema de reconocimiento de dígitos escritos a mano (Isabelle Guyon 2004). Este conjunto de datos forma parte de los cinco conjuntos utilizados en el desafío de selección de características NIPS 2003. Tiene 13500 observaciones y 5000 atributos (13500x5000), de los cuales solo 2500 son relevantes. El desafío radica en diferenciar los dígitos '4' y '9', que suelen ser fácilmente confundibles entre sí. Los dígitos han sido normalizados en tamaño y centrados en una imagen fija de 28x28 píxeles. Además, se crearon nuevas características como combinación de las existentes para construir un espacio de mayor dimensión. También se añadieron características distractoras denominadas "sondas", que no tienen poder predictivo. El orden de las características y patrones fue aleatorizado. Los datos están divididos en un conjunto de entrenamiento y un conjunto de testeo.

En este caso, nos encontramos en un escenario similar al de Madelon, con un dataset ruidoso e información redundante. La particularidad de Gisette es que, pese a mantener una relación positiva entre obervaciones y características (las primeras son más que las segundas), posee un espacio de búsqueda sensiblemente más grande y, eventualmente, más complejo que el de Madelon. Por esa razón, esperamos que este dataset sea computacionalmente más exigente que el anterior.

El dataset Leukemia es un análisis de datos de expresión genética obtenidos de microarreglos de ADN para la clasificación de tipos de cáncer (Golub et al. 1999). Se construyó un conjunto de datos con 72 observaciones y 7129 mediciones (72x7129) de las clases ALL (leucemia linfocítica aguda) y AML (leucemia mielogénica aguda). El problema es distinguir entre estas dos variantes de leucemia (ALL y AML). Los datos se dividen originalmente en dos subconjuntos: un conjunto de entrenamiento de 38 observaciones y un conjunto de testeo de 34 observaciones.

Con este dataset nos encontramos, precisamente, en el escenario de alta dimensionalidad y escasez muestral. El número de observaciones es menor que el de características, y las dimensiones del problema son significativamente más altas que en los casos anteriores. Además, el dataset está desbalanceado, con 27 observaciones de la clase ALL y 11 de la clase AML en la partición de entrenamiento, lo que plantea un desafío adicional para los algoritmos de aprendizaje.

El dataset ALL Leukemia es un estudio de pacientes pediátricos con leucemia linfoblástica aguda (LLA). Incluye 327 muestras con información de 12600 genes (327x12600). Está compuesto por 14 clases desequilibradas. Fue compilado por Yeoh et al. y está disponible para consulta[3].

Este caso es, sin duda, uno de los más complejos de todos, y plantea un desafío computacional significativo. El número de observaciones es significativamente menor que el de características, y las dimensiones del problema son significativamente más altas que en los casos anteriores. Además, el dataset está desbalanceado, con multiples clases desequilibradas, planteando no solo el desafío de las diversas fronteras de decisión

[3] Acceso público.

que se deben encontrar, sino también el problema de posibles ambigüedades entre clases. Como vimos en el capítulo anterior, el desafío de procesar información genética es significativo considerando la alta correlación entre genes, y la gran cantidad de información redundante.

Finalmente, el dataset GCM fue compilado en (Ramaswamy et al. 2001) y contiene los perfiles de expresión de muestras de tumores que representan 14 clases comunes de cáncer humano. El dataset está compuesto por 190 muestras y 16063 atributos (biomarcadores), distribuidos en clases desequilibradas. Los datos están divididos en un conjunto de entrenamiento y un conjunto de testeo. En GCM, como en el caso de ALL Leukemia, nos encontramos en un escenario de alta dimensionalidad y escasez muestral. El número de observaciones es significativamente menor que el de características, y las dimensiones del problema son significativamente más altas que en todos los casos anteriores.

Entendemos que esta variedad de datasets cubre un amplio espectro de problemas de aprendizaje, y que los resultados obtenidos en cada uno de ellos nos permitirán evaluar el desempeño de nuestros algoritmos en contextos distintos. Cabe aclarar que las particiones originales de los datasets fueron concatenadas en un solo conjunto de datos, y luego se dividió en conjuntos de entrenamiento y testeo en proporción 80/20.

## 2.2. El desempeño de algoritmos clásicos

Pasando a la evaluación de los modelos clásicos, hemos seleccionado una serie de modelos ampliamente usados en el campo del aprendizaje automático para tomar su desempeño como indicador. Entre ellos, encontramos: modelos lineales, modelos basados en árboles, modelos de Naive Bayes, modelos de vecinos más cercanos, modelos de redes neuronales y modelos de Máquinas de Soporte Vectorial. Cuando fue posible dada la naturaleza y características de los datasets, hemos evaluado todos los modelos. En caso contrario, hemos seleccionado aquellos modelos más apropiados para el contexto, dejando de lado los que no resultaban adecuados para el problema (como es el caso de GCM y Gisette, dada la dimensionalidad de ambos, y la naturaleza multiclase del primero). Finalmente, a fin de estandarizar la implementación de estos algoritmos, hemos empleado la librería `scikit-learn` que provee abstracciones convenientes para nuestro entorno de experimentación.

Los modelos lineales se basan en la premisa de que la variable objetivo puede expresarse como una función lineal de los predictores o características. Normalmente asumen que, para cada observación, la suma ponderada de las características (potencialmente con un término independiente o "bias") determina la respuesta. Entre los modelos lineales que incluimos en nuestro estudio se encuentran: el Análisis Discriminante Lineal (LDA), el Análisis Discriminante Cuadrático (QDA), la Regresión de Cresta (Ridge), y el Descenso de Gradiente Estocástico (SGD) (Hastie, Tibshirani, and Friedman 2009). El Análisis Discriminante Lineal (LDA) asume que cada clase proviene de una distribución normal multivariada con igual matriz de covarianzas y distinta media, y busca el hiperplano que maximiza la separabilidad entre clases proyectando los datos a un subespacio. El Análisis Discriminante Cuadrático (QDA) relaja el supuesto de matriz de covarianzas compartida, permitiendo que cada clase tenga su propia matriz y mejorando así la capacidad de modelar fronteras de decisión más complejas, aunque con un mayor riesgo de sobreajuste cuando se cuenta con poca muestra. La Regresión de Cresta (Ridge) introduce una penalización L2 sobre los coeficientes para controlar la varianza de la solución y mitigar la multicolinealidad,

lo cual es especialmente útil si existe una gran correlación entre características. El Descenso de Gradiente Estocástico (SGD), por su parte, consiste en un procedimiento incremental de optimización que actualiza los parámetros de un modelo lineal luego de cada observación (o mini-batch), resultando muy eficiente en problemas de alta dimensión o grandes volúmenes de datos.

En el caso de los modelos basados en árboles, todos comparten la idea de ir dividiendo recursivamente el espacio de características en regiones homogéneas. Este proceso se materializa en un árbol de decisión que, en cada nodo, escoge un umbral o criterio de partición para una característica. Los modelos incluidos en nuestro estudio son: Arbol de decisión clásico (DTC), AdaBoost, Bagging, Extra Trees Ensemble, Gradient Boosting, Random Forest, ETC, y Arboles Extremadamente Aleatorizados (ETC) (Hastie, Tibshirani, and Friedman 2009). El Árbol de Decisión Clásico (Decision Tree Classifier, DTC) utiliza criterios como la ganancia de información o la reducción de la impureza para decidir la partición óptima en cada nivel, siendo fácilmente interpretable aunque con tendencia al sobreajuste si no se regula su profundidad. Algunas variantes se basan en la combinación de múltiples árboles. Bagging, por ejemplo, entrena árboles independientes a partir de muestras "bootstrap"[4] y agrega las predicciones para reducir la varianza del modelo. Random Forest amplía esta idea, incorporando además la selección aleatoria de características en cada división, con lo cual reduce la correlación entre árboles y mejora la capacidad generalizadora. Extra Trees Ensemble adopta una estrategia aún más aleatoria, ya que define umbrales de corte aleatorios, lo que tiende a una mayor diversidad entre árboles y puede favorecer la reducción de la varianza. AdaBoost, en contraposición, entrena secuencialmente modelos débiles (a menudo árboles de baja profundidad) poniendo más peso en las observaciones mal clasificadas en iteraciones previas, de modo que cada nuevo modelo aprenda de los errores acumulados. Gradient Boosting también combina modelos débiles, pero en su caso cada etapa del entrenamiento se orienta a predecir el error residual del ensamble previo, optimizando una función de costo de forma aditiva y generalmente logrando modelos muy potentes. Cuando se habla de ETC (Extremely Randomized Trees Classifier), se hace referencia a un enfoque similar a Random Forest, pero que enfatiza la aleatorización tanto en la selección de subconjuntos de características como en los umbrales de partición, mejorando la diversidad de los árboles y mitigando así la varianza global.

Los modelos de Naive Bayes se basan en el teorema de Bayes para predecir la probabilidad de pertenencia a cada clase, asumiendo independencia condicional de las características (Hastie, Tibshirani, and Friedman 2009). Aun cuando esta suposición rara vez se cumple por completo en problemas reales, la simplicidad computacional y la eficacia empírica suelen convertirlos en una elección sólida, especialmente en problemas de alta dimensión. En su versión Bernoulli (BNB), se asume que las variables predictoras son binarias, lo que se ajusta bien a datos dispersos donde cada característica indica la presencia o ausencia de cierta propiedad. En la versión Gaussiana (GNB), se asume que cada característica sigue una distribución normal, estimando media y varianza por clase para luego combinar esas estimaciones en la regla de decisión bayesiana.

En cuanto a los métodos basados en vecinos más cercanos, el clasificador K-Vecinos Más Cercanos (KNN) ejemplifica la estrategia de aprendizaje por vecindad, ya que no construye un modelo explícito durante la etapa de entrenamiento. En su lugar, para

[4] El bootstrap es una técnica de remuestreo que consiste en generar múltiples conjuntos de datos mediante el muestreo aleatorio con reemplazo del conjunto original.

clasificar una nueva observación, identifica los K vecinos más cercanos en el espacio de características y asigna la clase mayoritaria de ese entorno local (Hastie, Tibshirani, and Friedman 2009). Este enfoque es intuitivo y puede capturar relaciones complejas en los datos, aunque su desempeño se degrada en alta dimensión y requiere un costo computacional alto en predicción, pues debe calcular distancias a todos los puntos de entrenamiento.

Entre los modelos de redes neuronales, el Perceptrón Multicapa (MLP) es una arquitectura de red con múltiples capas densamente conectadas y funciones de activación no lineales (Hastie, Tibshirani, and Friedman 2009). Su capacidad de aproximar funciones complejas lo convierte en un modelo flexible, pero también más exigente en términos de datos y calibración de hiperparámetros. Aunque en su versión más simple puede considerarse un "clásico", el MLP con técnicas de regularización y optimización robustas forma parte fundamental de las estrategias de aprendizaje profundo.

Por último, las Máquinas de Soporte Vectorial (SVM) parten del principio de encontrar un hiperplano (en el caso lineal) u "frontera" (en el caso con núcleos no lineales) que maximice el margen de separación entre clases (Hastie, Tibshirani, and Friedman 2009). En contextos de alta dimensión y con un número moderado de muestras, las SVM suelen mostrar un desempeño notable por su capacidad para controlar el sobreajuste mediante el parámetro de regularización y el uso de núcleos apropiados. Su versión lineal (LSVC) se centra en resolver una optimización con un límite que separa las clases en un espacio original de altas dimensiones sin necesidad de mapeos adicionales, resultando eficiente en muchos casos de datos dispersos. La variante NuSVC introduce un parámetro *nu* que controla tanto el número de vectores de soporte como la proporción máxima de errores permitidos, proporcionando una forma alternativa de regularización y definición de la frontera de decisión. Estas particularidades hacen que las SVM sean especialmente populares en problemas donde la dimensionalidad de las características es grande con respecto al número de muestras disponibles.

### 2.2.1. Configuración de los Modelos

Para evaluar el desempeño de los modelos clásicos y determinar la configuración más adecuada de sus hiperparámetros, hemos implementado un proceso de búsqueda sistemática en grilla (Grid Search). Este procedimiento consiste en seleccionar un conjunto de valores relevantes para cada hiperparámetro e iterar sobre todas las combinaciones posibles, entrenando y validando el modelo en cada caso.

En particular, establecimos un rango de parámetros numéricos que incluye de 3 a 20 valores, dependiendo de la sensibilidad del modelo a cada hiperparámetro y de los límites sugeridos en la literatura. Por ejemplo, en modelos lineales como Ridge o SGD, la regularización (penalty) y la tasa de aprendizaje (learning rate) se probaron en al menos tres niveles para capturar comportamientos distintos. En cambio, en algoritmos basados en árboles (como Random Forest o Gradient Boosting), ampliamos el rango hasta 20 valores en parámetros críticos (número de árboles, profundidad máxima, etc.) para reflejar la diversidad de configuraciones posibles.

Para los parámetros no numéricos (como funciones de activación en MLP, criterios de partición en árboles, tipo de kernel en SVM, entre otros) se utilizaron configuraciones estándar y reconocidas por la comunidad, con el fin de reducir la complejidad combinatoria. Aun así, para cada modelo se revisó la documentación de `scikit-learn` y la literatura relacionada, asegurando una cobertura apropiada de las variantes más importantes.

La métrica de evaluación que hemos seleccionado es AUC para los dataset de clasificación binaria (Madelon, Gisette y Leukemia), y macro-F1 para los dataset de clasificación multiclase (GCM).

La métrica AUC mide el área bajo la curva ROC, donde la curva ROC traza la Tasa de Verdaderos Positivos (TPR o Sensibilidad) frente a la Tasa de Falsos Positivos (FPR) a distintos umbrales de decisión. Matemáticamente, la AUC puede interpretarse (bajo ciertas condiciones) como la probabilidad de que el clasificador asigne una puntuación más alta a una muestra positiva que a una negativa. Un valor de AUC igual a 1 indica un modelo perfecto, mientras que un valor cercano a 0.5 sugiere un desempeño cercano al de un clasificador aleatorio.

En clasificación binaria, la AUC ofrece un panorama amplio del desempeño del modelo al no depender de un umbral específico y tiene en cuenta la relación entre verdaderos positivos y falsos positivos. Esto es especialmente relevante en datasets como Madelon, Gisette o Leukemia, donde el desbalance y la alta dimensionalidad pueden sesgar otras métricas.

La métrica F1 se define como la media armónica entre la Precisión (exactitud) y el Recall (sensibilidad): F1 = 2 × (Precisión × Recall) / (Precisión + Recall). La Precisión indica qué proporción de las predicciones positivas son realmente positivas, mientras que el Recall mide la proporción de positivos correctamente identificados respecto de todos los positivos reales. La F1 combina ambas, penalizando los modelos que desequilibran excesivamente la Precisión y el Recall. En problemas multiclase (por ejemplo, en GCM con 14 clases), la F1 puede calcularse por clase y luego promediarse (macro-F1) para evaluar la capacidad de cada modelo de reconocer de forma equilibrada todas las clases, incluso cuando existe desbalance.

En clasificación multiclase, la macro-F1 proporciona una forma de resumir la Precisión y el Recall cuando hay varias clases involucradas y potencialmente desbalanceadas. La macro-F1 balancea correctamente los falsos positivos y los falsos negativos para cada clase antes de promediar, lo que resulta más informativo para problemas con muchas clases desiguales (como GCM).

## 2.3. Resultados obtenidos

Las métricas presentadas en la Tabla 2.1, evidencian que los resultados de los modelos clásicos varían ampliamente según las características de cada dataset. En particular, los conjuntos de datos Leukemia y Gisette muestran, en términos generales, un desempeño sólido para la mayoría de los algoritmos analizados. La AUC alcanzada por varios modelos en Leukemia (en algunos casos 1.00) ilustra la capacidad de separar correctamente los casos, a pesar de tratarse de un problema con alta dimensionalidad y desbalance. Asimismo, Gisette se beneficia del hecho de contar con más observaciones que features, lo que facilita la labor de los clasificadores (algunos, como Gradient Boosting, alcanzan AUC = 1.00).

En contraste, Madelon y, muy especialmente, GCM, exhiben una dificultad sustancialmente mayor para casi todas las familias de modelos clásicos. En Madelon, la presencia de ruido y características irrelevantes afecta la capacidad de generalización, quedando reflejado en valores de AUC relativamente bajos (en general por debajo de 0.70). Esta situación concuerda con la naturaleza artificial de Madelon, donde únicamente un subconjunto pequeño de atributos tiene poder predictivo y los modelos se enfrentan a espacios de búsqueda con muchos distractores. Este resultado subraya la

relevancia de algoritmos que incorporen estrategias que mitiguen el impacto del ruido en el aprendizaje o mecanismos de selección de características que permitan descartar las irrelevantes.

Tab.2.1 Resultados en el dataset de testeo

| Modelo | Leukemia(auc) | Madelon(auc) | Gisette(auc) | GCM(mf1) |
|---|---|---|---|---|
| Modelos Lineales | | | | |
| LDA | 0.85 | 0.60 | 0.96 | - |
| QDA | 0.50 | 0.66 | 0.70 | - |
| Ridge | 0.99 | 0.60 | 0.97 | - |
| SGD | 0.98 | 0.64 | 0.99 | 0.71 |
| Árboles | | | | |
| AdaBoost | 0.91 | 0.84 | 0.99 | - |
| Bagging | 1.00 | 0.91 | - | - |
| DTC | 0.72 | 0.64 | 0.92 | 0.53 |
| ETC | 0.54 | 0.57 | 0.94 | 0.48 |
| Ext.Trees.Ens. | 1.00 | 0.71 | 0.99 | 0.57 |
| Gradient Boost. | 0.99 | 0.82 | 1.00 | 0.58 |
| Random Forest | 1.00 | 0.78 | 0.99 | 0.62 |
| SVM | | | | |
| LSVC | 0.99 | 0.62 | 0.99 | 0.62 |
| NuSVC | 1.00 | 0.61 | 0.99 | 0.58 |
| SVC | 1.00 | 0.61 | 0.99 | 0.58 |
| Naive Bayes | | | | |
| BNB | 0.89 | 0.63 | 0.94 | - |
| GNB | 0.91 | 0.65 | 0.85 | - |
| KNN | | | | |
| KNN | 0.86 | 0.65 | 0.99 | - |
| Redes Neuronales | | | | |
| MLP | 0.96 | 0.58 | 0.99 | 0.68 |

El caso de GCM es aún más crítico, como se puede apreciar en la Tabla 2.2. Se trata de un problema multiclase severamente desbalanceado y con una dimensionalidad desproporcionadamente alta en comparación con el número de muestras disponibles. Dichas condiciones propician que todos los modelos clásicos evidencien dificultades para capturar las fronteras de decisión y generalizar correctamente. El hecho de que el MLP sea, en este caso, el método de mayor desempeño (si bien con un F1 todavía moderado) puede atribuirse a la flexibilidad de las redes neuronales y su capacidad de aproximar funciones complejas, a pesar del reducido número de muestras de entrenamiento. No obstante, la mayoría de clasificadores enfrenta limitaciones para generalizar en este escenario de desequilibrio tan marcado, reforzando la hipótesis central de la tesis sobre la importancia de enfoques generativos para rebalancear la asimetría entre clases y mejorar la clasificación.

Obsérvese que ciertas clases con muy pocas muestras (por ejemplo, Breast o Prostate) presentan métricas nulas, mientras que otras con más muestras, como Leukemia o Lymphoma, muestran resultados más consistentes. La presencia de clases con tan baja representatividad hace que el modelo no cuente con suficiente evidencia estadística para aprender patrones característicos, conduciendo a predicciones erróneas (o directamente inexistentes) para dichas clases. Como anticipamos en el capítulo

anterior, en un escenario multiclase desbalanceado, puede ser común que las clases minoritarias obtengan poco o ningún peso en la función de pérdida, lo que lleva al modelo a ignorarlas o predecir muy pocas (o ninguna) instancias de dichas clases.

Los resultados que el MLP obtiene por clase se muestran en la Tabla 2.2.

Tab.2.2 Resultados por clase en GCM

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Bladder | 0.75 | 1.00 | 0.857 | 3 |
| Breast | 0.00 | 0.00 | 0.000 | 3 |
| CNS | 0.80 | 1.00 | 0.889 | 4 |
| Colorectal | 0.75 | 1.00 | 0.857 | 3 |
| Leukemia | 1.00 | 0.83 | 0.909 | 6 |
| Lung | 1.00 | 0.33 | 0.500 | 3 |
| Lymphoma | 0.83 | 0.83 | 0.833 | 6 |
| Melanoma | 1.00 | 0.50 | 0.667 | 2 |
| Mesothelioma | 1.00 | 1.00 | 1.000 | 3 |
| Ovary | 0.50 | 0.67 | 0.571 | 3 |
| Pancreas | 0.50 | 0.67 | 0.571 | 3 |
| Prostate | 0.00 | 0.00 | 0.000 | 2 |
| Renal | 1.00 | 0.33 | 0.500 | 3 |
| Uterus_Adeno | 0.40 | 1.00 | 0.571 | 2 |
| Prom. macro | 0.68 | 0.65 | 0.62 | 46 |
| Prom.ponderado | 0.73 | 0.70 | 0.68 | 46 |
| Exactitud[5] | | | 0.70 | 46 |

En consecuencia, en problemas donde la alta dimensionalidad y bajo número de muestras se combina con múltiples clases desequilibradas, aun los modelos con capacidad de generalización elevada (por ejemplo, redes neuronales) se ven limitados. La disparidad de resultados por clase que hemos reportado (desde métricas perfectas hasta valores nulos) ilustra la necesidad de una estrategia que aborde el desbalance de manera explícita, ya sea mediante incremento sintético de datos, la implementación de funciones de pérdida adaptativas (que penalicen más los errores en las clases minoritarias), o ambos.

En síntesis, la combinación de alta dimensionalidad, escasez muestral y desbalance de clases constituye un serio obstáculo al entrenamiento estable y robusto de modelos clásicos, tal como se constata en GCM. Por otro lado, incluso en problemas más sencillos como Madelon, el ruido y la alta proporción de atributos irrelevantes pueden perjudicar significativamente la performance de estos algoritmos. Estos hallazgos abonan la necesidad e importancia de la estrategia que exploraremos en este trabajo, que se centra en la generación de datos sintéticos para incrementar la muestra de entrenamiento y mejorar la capacidad de generalización de los modelos.

[5]Remitimos al lector al punto al punto 3.7 para una explicación detallada de las métricas presentadas aquí y a lo largo de nuestros experimentos.

Capítulo 3

# Autocodificadores Variacionales y datos sintéticos

En este capítulo presentamos la arquitectura del Autocodificador Variacional (AV) que emplearemos para la generación de datos sintéticos: exponemos brevemente sus fundamentos teóricos, los pasos que hemos seguido en su implementación y las variaciones introducidas para su apropiada aplicación a los problemas abordados. También presentamos las métricas empleadas para evaluar los modelos y las configuraciones más relevantes para la generación de datos sintéticos. Finalmente, concluimos el capítulo comentando los hallazgos más destacados de todo este desarrollo.

## 3.1. Modelos generativos

Los modelos generativos (MG) son un amplio conjunto de algoritmos de aprendizaje automático que buscan modelar la distribución de probabilidad de un conjunto de datos $p_\theta(x)$. A diferencia de los modelos discriminantes (MD), cuyo objetivo es aprender un predictor a partir de los datos, en los modelos generativos el objetivo es resolver un problema más general vinculado con el aprendizaje de la distribución de probabilidad conjunta de todas las variables. Así, siguiendo a Kingma, podemos decir que un modelo generativo simula la forma en que los datos son generados en el mundo real (Kingma and Welling 2019). Considerando estas propiedades, los modelos generativos permiten crear nuevos datos que se asemejan a los originales, y se aplican en tareas de generación de datos sintéticos, imputación de datos faltantes, reducción de dimensionalidad y selección de características, entre otros.

Los modelos generativos pueden tener como inputs diferentes tipos de datos, como imágenes, textos, audios, información tabular, etc. Por ejemplo, las imágenes son un tipo de dato para los cuales los MG han demostrado gran efectividad. En este caso, cada dato de entrada $x$ es una imagen que puede estar representada por un vector de miles de elementos que corresponden a los valores de píxeles. El objetivo de un modelo generativo es aprender las dependencias (Doersch 2021) entre los píxeles (e.g. pixeles vecinos tienden a tener valores similares) y poder generar nuevas imágenes que se asemejen a las imágenes originales.

Podemos formalizar esta idea asumiendo que tenemos ejemplos de datos $x$, distribuidos según una distribución de probabilidad conjunta no conocida que queremos modelar con $p_\theta(x)$ para que sea capaz de generar datos similares a los originales.

## 3.2. Autocodificadores

Los autocodificadores son un tipo de MG especializado en representar un espacio de características dado en un espacio de menor dimensión (de la Torre 2023). El objetivo de esta transformación es obtener una representación de baja dimensionalidad que preserve con la mayor fidelidad posible las propiedades del espacio original. Para ello el modelo aprende a preservar la mayor cantidad de información relevante en un vector denso de menos dimensiones que las originales, y descarta -al mismo tiempo- lo irrelevante. Luego, a partir de esa información codificada, se busca reconstruir los datos observados según el espacio original.

Los autocodificadores se componen de dos partes: un codificador y un decodificador. El codificador es una función no lineal que opera sobre una observación o patrón $x_i$ y la transforma en un vector de menor dimensión $z$, mientras que el decodificador opera a partir del vector $z$ y lo transforma en una observación $\hat{x_i}$ (o patrón reconstruido), buscando que se asemeje a la observación original. Este vector de menor dimensión $z$ es conocido como espacio latente.

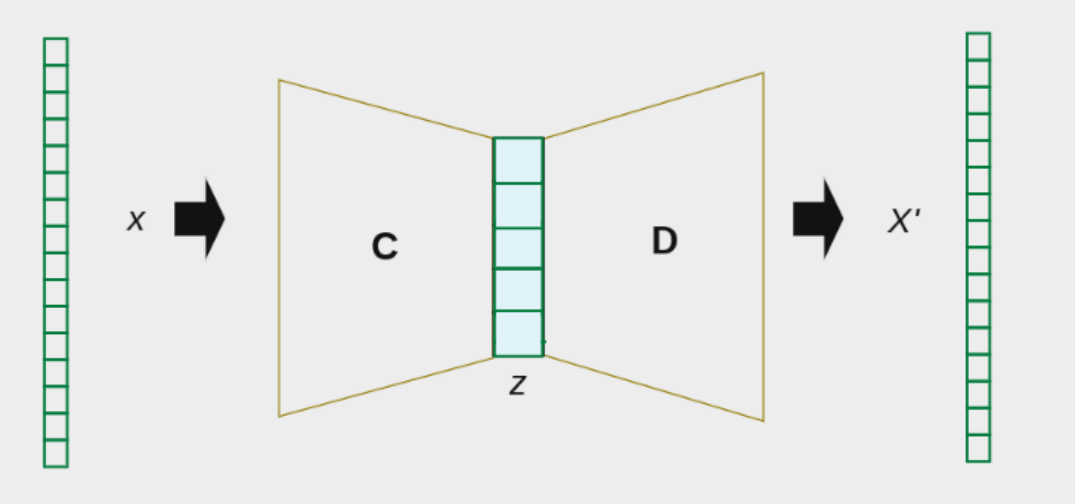


Figura 3.1 Ejemplo esquemático de un autocodificador

El esquema anterior muestra un autocodificador que transforma un patrón $x$ en un vector denso $z$, con dimensiones menores a las originales. Este vector es usado luego por el decodificador para reconstruir el patrón original, dando lugar a un patrón reconstruido $x'$.

En el proceso de aprendizaje de un autocodificador, la red modela la distribución de probabilidad de los datos de entrada $x$ y aprende a mapearlos a un espacio latente $z$. Para ello, se busca minimizar la diferencia entre la observación original $x_i$ y la reconstrucción $\hat{x_i}$, diferencia que se denomina error de reconstrucción. Esta optimización se realiza a través de una función de pérdida, que permite la optimización simultánea del codificador y decodificador.

Formalmente, podemos establecer estas definiciones (de la Torre 2023):

- Sea $x$ el espacio de características de los datos de entrada y $z$ el espacio latente, ambos espacios son euclidianos, $x = \mathbb{R}^m$ y $z = \mathbb{R}^n$, donde $m > n$.
- Sea las siguientes funciones paramétricas $C_\theta : x \to z$ y $D_\phi : z \to x'$ que representan el codificador y decodificador respectivamente.

- Entonces para cada observación $x_i \in x$, el autocodificador busca minimizar la función de pérdida $L(x_i, D_\phi(E_\theta(x_i)))$. Ambas funciones $E_\theta$ y $D_\phi$ son redes neuronales que se entrenan simultáneamente.

Para optimizar un autocodificador se requiere una función que permita medir la diferencia entre la observación original y la reconstrucción. Esta diferencia usualmente se basa en la distancia euclidia entre $x_i$ y $\hat{x}_i$, es decir, $||x_i - x_i'||^2$. La función de pérdida se define como la suma de todas las distancias a lo largo del conjunto de datos de entrenamiento. Tenemos entonces que:

$$L(\theta, \phi) = argmin_{\theta,\phi} \sum_{i=1}^{N} ||x_i - D_\phi(C_\theta(x_i))||^2, \tag{3.1}$$

donde $L(\theta, \phi)$ representa la función de pérdida que queremos minimizar: $\theta$ son los parámetros del codificador $C$ y $\phi$ son los parámetros del decodificados $D$.

## 3.3. Autocodificadores y el problema de la generación de datos

En el proceso de aprendizaje antes descripto, la optimización no está sujeta a otra restricción mas que minimizar la diferencia entre la observación original y la reconstrucción, dando lugar a espacios latentes generalmente discontinuos. Esto sucede porque la red puede aprender a representar los datos de entrada de manera eficiente sin necesidad de aprender una representación continua. En la arquitectura del autocodificador no hay determinantes para que dos puntos cercanos en el espacio de características se mapeen a puntos cercanos en el espacio latente.

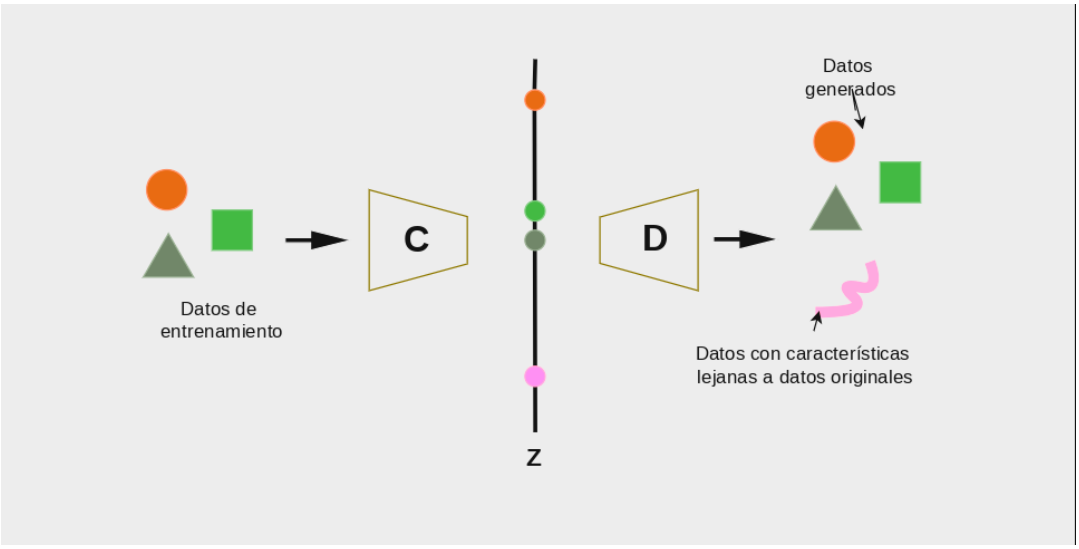


Figura 3.2 Discontinuidad del espacio latente

Esta discontinuidad en el espacio latente hace posible que ciertas regiones de este espacio no tengan relación significativa con el espacio de características. Durante el entrenamiento el modelo simplemente no ha tenido que reconstruir datos cuyas distribuciones coincidan con estas regiones. Esto es un problema en la generación de datos, ya que la red podrá generar representaciones alejadas de los datos originales. Regularmente lo que se busca en los MG, no es simplemente una generación de datos completemante igual o totalmente distintos a los orginales, sino cierta situación intermedia donde los nuevos datos introducen variaciones en características específicas.

Un MG está conformado por dos componentes principales, un codificador y un decodificador. El vector de entrada continuo se somete a una transformación no lineal en el codificador, obteniendo así la representación latente en un espacio de menor dimensión. Esta transformación comprime la información original para capturar sus características más relevantes, mientras que el decodificador se encarga de proyectar la representación resultante de nuevo al espacio de características, generando una salida que se aproxima a la muestra inicial. Sin embargo, ciertas regiones de dicho espacio pueden quedar sin representar, y al muestrear en esas zonas, el modelo puede producir instancias muy alejadas de las distribuciones originales, como figuras que no 'encajan' entre aquellas que aparecen en el espacio de características, tal como se ilustra en la Figura 3.2. Este fenómeno está directamente relacionado con la discontinuidad del espacio latente, y pone de manifiesto la necesidad de enfoques que incorporen regularizaciones o mecanismos adicionales para lograr una generación más coherente.

## 3.4. Autocodificadores Variacionales

Los Autocodificadores Variacionales (AV) buscan resolver los problemas de discontinuidad y falta de regularidad en el espacio latente de los Autocodificadores. Comparten con éstos la arquitectura codificador-decodificador, pero introducen importantes modificaciones en su diseño para crear un espacio latente continuo.

Estos modelos, a diferencia de los autocodificadores que realizan transformaciones determinísticas de los datos de entrada (codificándolos como vectores n-dimensionales), buscan modelar la distribución de probabilidad de dichos datos aproximando la distribución a posteriori de las variables latentes $p_\theta(z|x)$. Para ello, la codificación se produce mediante la generación de dos vectores $\mu$ y $\sigma$ (vector de medias y vector de desvíos estándar) que conforman el espacio latente, a partir del cual se toman las muestras para la generación.

La red codificadora, también llamada red de reconocimiento, mapea los datos de entrada $x$ a los vectores $\mu$ y $\sigma$, que parametrizan una distribución de probabilidad en el espacio latente. Generalmente, esta distribución es una distribución simple, como la distribución normal multivariada. La red decodificadora, también llamada red generativa, toma muestras de esta distribución para generar un vector, y lo transforma según la distribución de probabilidad preexistente del espacio de características. De esta manera, se generan nuevas instancias que respetan la distribución de probabilidad de los datos originales. Estas transformaciones implican que, incluso para el mismo dato observado (donde los parámetros de $z$ son iguales), el dato de salida podrá ser diferente debido al proceso estocástico de reconstrucción.

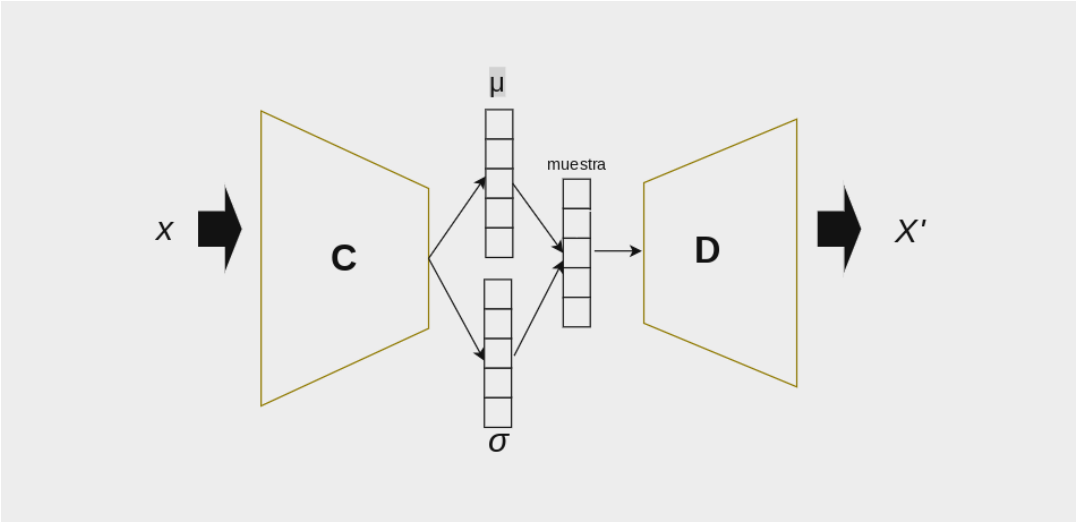


Figura 3.3 Autocoficadores Variacionales

Una forma de entender esta arquitectura sería relacionar los vectores que componen $z$ como 'referecias', donde el vector de medias controla el centro en torno al cual se distribuirán los valores codificados de los datos de entrada, mientras que el vector de los desvíos traza el área que pueden asumir dichos valores en torno al centro.

Para indagar en estas intuiciones, veamos la solución que proponen los AV detenidamente, utilizando un enfoque formal. Así, dado un conjunto de datos de entrada $\mathbf{X} = \{x_1, x_2, ..., x_N\}$, donde $x_i \in \mathbb{R}^m$, se asume que cada muestra es generada por un mismo proceso o sistema subyacente cuya distribución de probabilidad se desconoce. El modelo buscado procura aprender $p_\theta(x)$, donde $\theta$ son los parámetros de la función. Por las ventajas que ofrece el logaritmo[1] para el cálculo de las distribuciones de probabilidad tendremos la siguiente expresión: $\log p_\theta(x) = \sum_{x_i \in x} \log p_\theta(x_i)$ [2].

La forma más común de calcular el parámetro $\theta$ es a través del estimador de máxima verosimilitud, cuya función de optimización es: $\theta^* = \arg\max_\theta \log p_\theta(x)$, es decir, buscamos los parámetros $\theta$ que maximizan la log-verosimilitud asignada a los datos por el modelo.

En el contexto de los AV, el objetivo es modelar la distribución de probabilidad de los datos observados $x$ a través de una distribución de probabilidad conjunta de variables observadas y latentes: $p_\theta(x, z)$. Aplicando la regla de la cadena de probabilidad podemos factorizar la distribución conjunta de la siguiente manera: $p_\theta(x, z) = p_\theta(x|z)p_\theta(z)$. Aquí $p_\theta(x|z)$ es la probabilidad condicional de los datos observados dados los latentes, y $p_\theta(z)$ es la probabilidad a priori[3] de los latentes.

Para determinar la distribución marginal respecto de los datos observados, es preciso integrar sobre todos los elementos de $z$, dando como resultado la siguiente función: $p_\theta(x) = \int p_\theta(x, z)dz$.

Esta distribución marginal puede ser extremadamente compleja, y contener un número indeterminable de dependencias (Kingma and Welling 2019), volviendo el calculo

[1] El logaritmo convierte la probabilidad conjunta (que se calcula como el producto de las probabilidades condicionales) en una suma de logaritmos, facilitando el cálculo y evitando problemas de precisión numérica: $\log(ab) = \log(a) + \log(b)$.

[2] Esta función se lee como la log-verosimilitud de los datos observados $x$ bajo el modelo $p_\theta(x)$ y es igual a la suma de la log-verosimilitud de cada dato de entrada $x_i$.

[3] La expresión a priori alude a que no está condicionada por ningun dato observado.

de la verosimilitud de los datos observados intratable. Esta intratabilidad de $p_\theta(x)$ está determinada por la intratabilidad de la distribución a posteriori $p_\theta(z|x)$, cuya dimensionalidad y multi-modalidad pueden hacer difícil cualquier solución analítica o numérica eficiente. Dicho obstáculo impide la diferenciación y por lo tanto la optimización de los parámetros del modelo.

Para abordar este problema, se acude a la inferencia variacional que introduce una aproximación $q_\phi(z|x)$ a la verdadera distribución a posteriori $p_\theta(z|x)$. Generalmente se emplea la distribución normal multivariada para aproximar la distribución a posteriori, con media y varianza parametrizadas por la red neuronal[4]. Sin embargo, la elección de la distribución no necesariamente tiene que pasar por una distribución normal, el único requerimiento es que sea una distribución que permita la diferenciación y el cálculo de la divergencia entre ambas distribuciones (por ejemplo si $X$ es binaria la distribución $p_\theta(x|z)$ puede ser una distribución Bernoulli).

Así, en lugar de maximizar directamente el logaritmo de la verosimilitud (log-verosimilitud), se maximiza una cota inferior conocida como límite inferior de evidencia (ELBO por sus siglas en ingles). La derivación procede de la siguiente manera:

1. log-verosimilitud marginal (intratable):

   $\log p_\theta(x) = \log\left(\int p_\theta(x,z)\,dz\right)$,

2. aplicando inferencia variacional:

   $\log p_\theta(x) = \log\left(\int q_\phi(z|x)\frac{p_\theta(x,z)}{q_\phi(z|x)}\,dz\right)$,

3. aplicando la desigualdad de Jensen[5]:

   $\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}\left[\log\left(\frac{p_\theta(x,z)}{q_\phi(z|x)}\right)\right]$,

4. descomponiendo la fracción dentro del logaritmo:

   $\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z) + \log p_\theta(z) - \log q_\phi(z|x)\right]$,

5. el resultando es el límite inferior de evidencia:

   $\log p_\theta(x) \geq \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z))$,

donde:

- $\mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)]$ es el valor esperado (esperanza) de la log-verosimilitud bajo la aproximación variacional, y determina la precision de la reconstrucción de los datos de entrada (un valor alto de esta esperanza indica que el modelo es capaz de reconstruir los datos de entrada con alta precisión a partir de los parámetros generados por $q_\phi(z|x)$).

[4]En AV, se suele asumir que $z$ sigue una distribución normal multivariada: $p_\theta(z) = \mathcal{N}(z; 0, I)$, con media cero y matriz de covarianza identidad. La matriz de covarianza identidad es una matriz diagonal con unos en la diagonal y ceros en los demás lugares, y su empleo simplifica la implementación del modelo, permite que las variables latentes sean independientes (covarianza = 0) y varianza unitaria, evitando así cualquier complejidad vinculada a las dependencias entre dimensiones de $z$.

[5]Nótese que ese límite es siempre menor o igual y esto se deriva de una de las propiedades de las funciones convexas. Esta propiedad, denominada desigualdad de Jensen, establece que el valor esperado de una función convexa es siempre mayor o igual a la función del valor esperado. Es decir, $\mathbb{E}[f(x)] \geq f(\mathbb{E}[x])$. En el caso de funciones cóncavas, la desigualdad se invierte: $\mathbb{E}[f(x)] \leq f(\mathbb{E}[x])$. En este caso, la función logaritmo es cóncava, por lo que la desigualdad se expresa como: $\log(\mathbb{E}[x]) \geq \mathbb{E}[\log(x)]$.

- $D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z))$ es la divergencia de Kullback-Leibler entre la distribución $q_\phi(z|x)$ y la distribución a priori de las variables latentes $p_\theta(z)$, y determina la regularización del espacio latente.

Maximizando esta cota inferior (ELBO), se optimizan simultáneamente los parámetros $\theta$ del modelo y los parámetros $\phi$ de la distribución empleada en la aproximación, permitiendo una inferencia eficiente y escalable en modelos con $z$ de alta dimensionalidad. En la teoría, cuando derivamos el objetivo de un AV, estamos maximizando la cota inferior variacional (ELBO), para que la aproximación sea lo más cercana posible a la verdadera distribución de los datos. LLevado el problema a una implementación práctica generalmente se emplean optimizadores (SGD, Adam, etc.) que minimizan una función de pérdida. Para convertir el problema de maximización del ELBO en un problema de minimización, simplemente negamos el ELBO, resultando que los términos de la ecuación se reescriben como suma de cantidades positivas. El error o pérdida de reconstrucción se mide, según los casos, mediante MSE o entropía cruzada.

El objetivo de aprendizaje del AV se da entonces por:

$$\mathcal{L}_{\theta,\phi}(x) = \max(\phi,\theta)\left(E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z))\right).$$

Como puede apreciarse en la ecuación anterior la función de pérdida del AV se compone de dos términos: el primero es la esperanza de la log-verosimilitud bajo la aproximación variacional y el segundo es la divergencia de Kullback-Leibler relacionada a la reconstrucción de los datos y la regularización del espacio latente. Existe entre ambos términos una relación de compromiso que permite al AV aprender una representación adecuada de los datos de entrada y, al mismo tiempo, un espacio latente continuo y regularizado.

En términos prácticos, la divergencia de Kullback-Leibler puede interpretarse como un factor que equilibra creatividad y precisión en el modelo. Por un lado, cuando esta divergencia es grande ($D_{\mathrm{KL}} \gg 1$), el espacio latente tiende a estar más regularizado y la distribución de probabilidad de los datos generados se vuelve más "suave", pero el modelo puede terminar sobreajustándose a las muestras originales y perder capacidad de generalización. Por otro lado, si la divergencia es muy pequeña ($D_{\mathrm{KL}} \ll 1$), el espacio latente queda menos regularizado y el modelo intenta reproducir con mayor fidelidad la distribución real de los datos de entrada, aunque a costa de generar muestras más ruidosas y posiblemente descuidar detalles esenciales. De esta manera, el valor de $D_{\mathrm{KL}}$ refleja el compromiso entre capturar información detallada de los datos y mantener una representación latente estable y bien estructurada para la generación.

## 3.5. Presentación de nuestros modelos de AV y AVC

Luego del análisis precedente exponemos a continuación los modelos de AV y AV Condicional (AVC) que desarrollamos para la generación de datos sintéticos. El desarrollo de dichos modelos se cumplió en dos etapas: en la primera se centró el esfuerzo en el diseño y validación de las arquitecturas más apropiadas para los problemas abordados, mientras que en la segunda se enfocó en su optimización. A continuación describiremos brevemente este proceso y la configuración final de los modelos elegidos para los experimentos de aumentación.

La primera etapa comenzó con la creación de una versión exploratoria de los modelos AV y AVC, con la finalidad de establecer una base sobre la cual iterar en mejoras

sucesivas. Se optó por redes de 2 capas en el codificador y en el decodificador en ambos casos, siguiendo la estructura descripta precedentemente.

El codificador incluyó dos capas lineales, cada una seguida de una activación ReLU, un diseño que sigue la lógica de la transformación no lineal en un espacio de menor dimensión para luego reconstruir la observación original a partir del espacio latente. Como se discutió en la primera parte, el codificador genera dos vectores, uno para la media y otro para la varianza logarítmica de la distribución latente, componentes críticos para el proceso de reparametrización que permite al modelo generar nuevas muestras en el espacio latente. El decodificador, encargado de reconstruir los datos originales a partir del espacio latente, fue diseñado con una estructura simétrica a la del codificador, utilizando nuevamente activaciones ReLU.

La función de pérdida de ambos modelos combinó la divergencia Kullback-Leibler y el error cuadrático medio. La $D_{KL}$ se utilizó para medir la diferencia entre la distribución aprendida por el modelo y una distribución normal estándar. El MSE y MSE balanceado para los problemas multiclases se emplearon para evaluar el error de reconstrucción, es decir, qué tan bien el modelo era capaz de replicar los datos de entrada a partir del espacio latente.

Cabe acalarar que, en escenarios orientados a la generación, consideramos particularmente apropiado emplear MSE para la función de reconstrucción, ya que permite aproximar cada dimensión de la muestra de manera continua. Métricas orientadas a la clasificación, como por ejemplo la entropía cruzada, penalizan con fuerza cuando la clase verdadera difiere de la predicha, mientras que la MSE interpreta la diferencia entre la muestra real y la reconstruida como una distancia. Entendemos que esta característica beneficia el proceso de entrenamiento porque se ajusta a la naturaleza continua de los datos generados y mantiene una penalización homogénea en cada dimensión.

Los modelos resultantes se probaron en la generación de datos sintéticos en un dataset de clases binarias: Madelon, y un dataset multiclases: GCM. Para evaluar los datos generados se realizaron experimentos de clasificación utilizando un Perceptron Multicapa (MLP), comparando los resultados obtenidos en el dataset original y en el dataset con muestras sintéticas. Los resultados de esta evaluación fueron satisfactorios, logrando una versión inicial de los modelos AV y AVC que permitían generar datos sintéticos. Sin perjuicio de ello, considerando el bajo desempeño del MLP sobre los datos sintéticos en comparación con el MLP sobre los datos originales, quedó en evidencia la baja calidad de reconstrucción que tenían ambos modelos. Circunstancia que nos llevó a buscar una arquitectura más adecuada.

Así, en respuesta a los problemas identificados previamente, se exploraron diferentes configuraciones de arquitectura para el AV y el AVC, determinándose que una configuración con tres capas lineales en el codificador y el decodificador, cada una con activaciones ReLU seguidas de normalización por lotes, brindaba los mejores resultados.

La técnica de normalización por lotes fue seleccionada debido a su capacidad para estabilizar y acelerar el proceso de entrenamiento, promoviendo la rápida convergencia y mejorando la precisión de la reconstrucción. Al mitigar el problema de desplazamiento de covariables (covariate shift) durante el entrenamiento, la normalización por lotes estabiliza las activaciones intermedias de la red y permite que la información relevante sea conservada a lo largo de las capas.

Dado que los modelos fueron ajustados para capturar la estructura de los datos a través de capas lineales y normalización por lotes, mantuvimos la elección de ReLU como función de activación dada su eficiencia computacional.

Los modelos resultantes fueron entrenados con un optimizador Adam y una tasa de aprendizaje en el rango de [1e-5, 1e-3]. Se experimentó con diferentes tamaños del espacio latente, evaluando el equilibrio entre la calidad de reconstrucción y la capacidad de generalización del modelo. Se empleó un término de paciencia para detener el entrenamiento si no se observaba mejora en los datos de validación durante 10 épocas consecutivas.

Estos experimentos fueron clave para ajustar los modelos generativos a las necesidades específicas de los conjuntos de datos utilizados en la investigación, permitiendo una generación de datos sintéticos que no solo replicara los patrones de los datos originales, sino que también capturara la variabilidad inherente a estos. Las arquitecturas finales de los modelos se detallan a continuación.

Arquitectura del Autocodificador Variacional

La arquitectura del AV está compuesta por tres capas lineales, cada una seguida de una normalización por lotes (Batch Normalization) y una activación ReLU. El proceso de codificación se realiza de la siguiente manera:

$h_1 = \text{ReLU}(\text{BatchNorm}(\mathbf{W_1}x + b_1))$,
$h_2 = \text{ReLU}(\text{BatchNorm}(\mathbf{W_2}h_1 + b_2))$,
$h_3 = \text{ReLU}(\text{BatchNorm}(\mathbf{W_3}h_2 + b_3))$,

donde:

- $\mathbf{W_1}$ es una matriz de pesos que transforma el vector de entrada $x$ al espacio de características de dimensión $|h_1|$.
- $\mathbf{W_2}$ transforma $h_1$ a un espacio de características de dimensión $|h_2|$.
- $\mathbf{W_3}$ transforma $h_2$ a un espacio de características de dimensión $|h_3|$.
- $\mathbf{b_1}$, $\mathbf{b_2}$, y $\mathbf{b_3}$ son los sesgos correspondientes a cada capa.

Después de las tres capas, se generan los vectores latentes $\mu$ y $\log(\sigma^2)$ mediante dos capas lineales independientes que no aplican normalización por lotes.

Reparametrización

El vector latente $z$ se obtiene mediante la técnica de reparametrización, donde se introduce ruido gaussiano para permitir la retropropagación del gradiente:

$z = \mu + \sigma \times \epsilon$,

donde $\epsilon$ es una variable aleatoria con distribución normal estándar, y $\sigma$ se calcula a partir de $\log(\sigma^2)$.

Decodificador

El decodificador reconstruye el vector de entrada a partir del vector latente $z$ utilizando una arquitectura de tres capas lineales, cada una seguida por una normalización por lotes y los dos primeras por una activación ReLU:

$h_4 = \text{ReLU}(\text{BatchNorm}(\mathbf{W_4}z + b_4))$,
$h_5 = \text{ReLU}(\text{BatchNorm}(\mathbf{W_5}h_4 + b_5))$,
$\hat{x} = \text{BatchNorm}(\mathbf{W_6}h_5 + b_6)$,

donde:

- $\mathbf{W_4}$ transforma el vector latente $z$ al espacio de características de dimensión $|h_4|$.
- $\mathbf{W_5}$ transforma $h_4$ al espacio de características de dimensión $|h_5|$.
- $\mathbf{W_6}$ transforma $h_5$ de regreso al espacio de la dimensión original de la entrada $|D_{in}|$.
- $\mathbf{b_4}$, $\mathbf{b_5}$, y $\mathbf{b_6}$ son los sesgos correspondientes a cada capa.

Finalmente, la salida $\hat{x}$ es una aproximación reconstruida de la entrada original $x$.

### 3.5.1. Modelo AVC para datos multiclase

Para abordar los dataset GCM y ALL Leukemia, que contienen multiples clases con distribuciones desiguales, se creó un AVC que combina la capacidad de generación de un AV tradicional con el condicionamiento explícito en las etiquetas de clase. El AVC propuesto permitió una modelización más precisa de los datos, al incorporar información de clase en el proceso de codificación y decodificación.

En escenarios donde los datasets están desbalanceados, los modelos generativos pueden tender a favorecer las clases mayoritarias, ignorando las minoritarias. Para abordar el desbalance de clases que presenta GCM, se implementó una estrategia de ponderación de clases dentro de la función de pérdida, penalizando de manera diferenciada los errores de reconstrucción en función de la clase, buscando mejorar así la capacidad del modelo para representar adecuadamente las clases minoritarias.

Arquitectura del Autocodificador Variacional Condicional

La arquitectura del AVC se basa en una modificación del AV tradicional para incorporar información adicional en forma de etiquetas. Esta información se concatena tanto en la fase de codificación como en la de decodificación, permitiendo que el modelo aprenda distribuciones condicionales.

Codificador

El codificador del AVC combina la entrada original con las etiquetas antes de ser procesadas por una secuencia de capas lineales (3 capas), cada una seguida por una normalización por lotes (Batch Normalization) y una activación ReLU. El proceso de codificación se realiza de la siguiente manera:

$h_1 = \text{ReLU}(\text{BatchNorm}(\mathbf{W_1}[x, y] + b_1))$,
$h_2 = \text{ReLU}(\text{BatchNorm}(\mathbf{W_2}h_1 + b_2))$,
$h_3 = \text{ReLU}(\text{BatchNorm}(\mathbf{W_3}h_2 + b_3))$,

donde:

- $[x, y]$ es la concatenación del vector de entrada $x$ con las etiquetas $y$.
- $\mathbf{W_1}$ es una matriz de pesos que transforma el vector combinado $[x, y]$ al espacio de características de dimensión $|h_1|$.
- $\mathbf{W_2}$ transforma $h_1$ a un espacio de características de dimensión $|h_2|$.
- $\mathbf{W_3}$ transforma $h_2$ a un espacio de características de dimensión $|h_3|$.
- $\mathbf{b_1}$, $\mathbf{b_2}$, y $\mathbf{b_3}$ son los sesgos correspondientes a cada capa.

Al igual que en el AVC, se generan los vectores latentes $\mu$ y $\log(\sigma^2)$ mediante dos capas lineales independientes.

Reparametrización

El vector latente $z$ se obtiene mediante la técnica de reparametrización, similar al AV tradicional:

$z = \mu + \sigma \times \epsilon$,

donde $\epsilon$ es una variable aleatoria con distribución normal estándar, y $\sigma$ se calcula a partir de $\log(\sigma^2)$.

Decodificador

El decodificador del AVC reconstruye el vector de entrada a partir del vector latente $z$ y las etiquetas $y$, utilizando una arquitectura de tres capas lineales, cada una seguida por una normalización por lotes y una activación ReLU:

$h_4 = \text{ReLU}(\text{BatchNorm}(\mathbf{W_4}[z, y] + b_4))$,
$h_5 = \text{ReLU}(\text{BatchNorm}(\mathbf{W_5} h_4 + b_5))$,
$\hat{x} = \text{BatchNorm}(\mathbf{W_6} h_5 + b_6)$,

donde:

- $[z, y]$ es la concatenación del vector latente $z$ con las etiquetas $y$.
- $\mathbf{W_4}$ transforma el vector combinado $[z, y]$ al espacio de características de dimensión $|h_4|$ + longitud de las categorías.
- $\mathbf{W_5}$ transforma $h_4$ al espacio de características de dimensión $|h_5|$.
- $\mathbf{W_6}$ transforma $h_5$ de regreso al espacio de la dimensión original de la entrada $|D_{in}|$.
- $\mathbf{b_4}$, $\mathbf{b_5}$, y $\mathbf{b_6}$ son los sesgos correspondientes a cada capa.

Finalmente, la salida $\hat{x}$ es una aproximación reconstruida de la entrada original $x$, condicionada por las etiquetas $y$.

Elementos distintivos respecto a la arquitectura anterior:

- Incorporación de etiquetas: Tanto en el codificador como en el decodificador, se concatenan las etiquetas $y$ con las entradas y el vector latente, respectivamente.
- Dimensiones ajustadas: Se han ajustado las dimensiones de las capas para incorporar las etiquetas, reflejadas en las matrices de pesos y las normalizaciones por lotes.
- Capas adicionales en la fase de decodificación: Se añaden capas y ajustes para manejar las etiquetas adicionales en el proceso de decodificación.

## 3.6. Parametrización de los modelos AV y AVC

La búsqueda y ajuste de hiperparámetros para los modelos de AV y AVC ha sido un proceso crucial para optimizar la generación de datos sintéticos. Asumíamos que este proceso era determinante para favorecer la estrategia de selección de características mediante el AG, por esa razón analizamos distintas opciones para la configuración de los modelos.

Iniciamos esta tarea realizando una búsqueda de hiperparámetros, ajustando aspectos clave como el tamaño de las dimensiones latentes, la tasa de aprendizaje, y el número de neuronas en las diferentes capas. Se utilizaron tanto Grid Search (Bergstra and Bengio, 2012) como Optimización Bayesiana (Snoek et all, 2012). Cada una de estas técnicas tiene sus fortalezas, y la elección entre ellas depende en gran medida del objetivo de la búsqueda. Grid Search, por ejemplo, permite un control total sobre el espacio de búsqueda, lo que es útil para responder preguntas específicas, como la configuración óptima de la dimensión latente. Sin embargo, la BO demostró ser particularmente eficiente en la exploración de un espacio de hiperparámetros más amplio y menos definido, logrando un equilibrio entre la exploración y la explotación que resultó especialmente útil en nuestra investigación dado el tamaño del espacio de búsqueda.

También exploramos técnicas para evitar el sobreajuste, como un mecanismo de paciencia para detener el entrenamiento si no se observaba mejora durante 10 épocas consecutivas en datos de validación. Además, se experimentó con el uso de dropout como técnica de regularización; se probaron tasas de dropout en un rango de 0.05 a 0.5 en distintas configuraciones de AVC, tanto con arquitecturas pequeñas (100-500 neuronas por capa) como grandes (1000-7000 neuronas por capa). Testeamos distintas métricas de pérdida, como L1 en lugar de MSE para evaluar si la L1_loss podría ofrecer mejoras.

Para abordar el problema del desbalance en conjuntos de datos multiclases, se implementaron ajustes específicos en la configuración del modelo y en las estrategias de entrenamiento. El primero consistió en la asignación de pesos diferenciados a las clases dentro de la función de pérdida, con el objetivo de aumentar la penalización por errores de clasificación en las clases minoritarias. Como vimos en el primer capítulo, el uso de pesos de clase es una técnica comúnmente utilizada para abordar este problema. Estos pesos se calcularon de manera inversamente proporcional a la frecuencia de las clases en el conjunto de datos, lo que significa que las clases con menos instancias recibieron pesos más altos. Al incorporar estos pesos de clase en la función de pérdida, nos aseguramos que los errores en las clases con menor representación tuvieran un impacto mayor durante el proceso de optimización, logrando un aprendizaje más equilibrado.

Además de ajustar la función de pérdida para mitigar el desbalance, se implementó una estrategia de muestreo ponderado en el proceso de entrenamiento. A través de un muestreador aleatorio ponderado, nos aseguramos que cada mini-lote de datos durante el entrenamiento tuviera una representación equilibrada de cada clase, asignando mayor probabilidad de ser seleccionadas a las instancias de clases minoritarias.

Para garantizar que la combinación de ambas estrategias no tuviera un efecto indeseado en el desempeño del modelo, se llevaron a cabo experimentos con el fin de validar la aplicación simultánea de ambas técnicas. Un efecto a evitar era, precisamente, que el modelo se volviera excesivamente sensible a las clases minoritarias, en detrimento de su capacidad para generalizar correctamente en clases mayoritarias. Los resultados de estos experimentos mostraron que, implementadas en conjunto, ambas estrategias lograban los mejores resultados.

## 3.7. Métricas empleadas para evaluar resultados

Para evaluar los resultados de nuestros modelos, siguiendo a Fajardo et al. (2021), comparamos las métricas alcanzadas por un MLP entrenado con datos sintéticos y evaluado con datos reales de testeo, con las métricas obtenidas por un MLP entrenado con datos reales y también evaluado en los mismos datos de testeo. En todos los casos se emplearon las particiones originales de los datos, a fin de propiciar la comparación de nuestros resultados con los obtenidos en la literatura. El indicador particular empleado para la comparación fue el de exactitud, que mide la proporción de predicciones correctas sobre el total de predicciones realizadas.

Además, se emplearon otras métricas para completar nuestro análisis, a saber: la precisión, el recall y el F1 Score. La precisión mide la proporción de predicciones positivas correctas sobre el total de predicciones positivas realizadas, es decir, qué tan preciso es el modelo al identificar casos positivos. El recall (o sensibilidad) indica la proporción de verdaderos positivos detectados sobre el total de positivos reales, reflejando la capacidad del modelo para capturar todos los casos positivos. El F1 Score es la media armónica de la precisión y el recall, proporcionando un equilibrio entre ambos y siendo especialmente útil cuando existe un desbalance en las clases. Otras métricas que utilizamos fueron el Promedio Macro y el Promedio Ponderado. Los indicadores de Promedio Macro calculan la media aritmética de las métricas individuales de cada clase sin considerar el número de muestras por clase, otorgando igual peso a todas las clases. Esto es útil para evaluar el rendimiento general en presencia de clases desbalanceadas. Por otro lado, el Promedio Ponderado tiene en cuenta el soporte (número de muestras) de cada clase al calcular la media, lo que significa que las clases con más muestras influyen más en el promedio general. Esto proporciona una visión más realista del rendimiento cuando hay clases dominantes.

## 3.8. Resultados obtenidos en los experimentos

Los resultados obtenidos en nuestros experimentos revelaron que un Perceptron Multicapa (MLP) entrenado con datos sintéticos generados por un AV/AVC puede igualar o incluso superar en ciertos casos el rendimiento de un MLP entrenado con datos reales. En la Figura 3.4 se muestran los resultados obtenidos en los experimentos realizados para los cuatro conjuntos de datos estudiados, donde comparamos datos sintéticos con datos reales: en azul se indican los resultados en los datos de testeo para el entrenamiento con datos reales, y en rojo los resultados de entrenar con datos sintéticos.

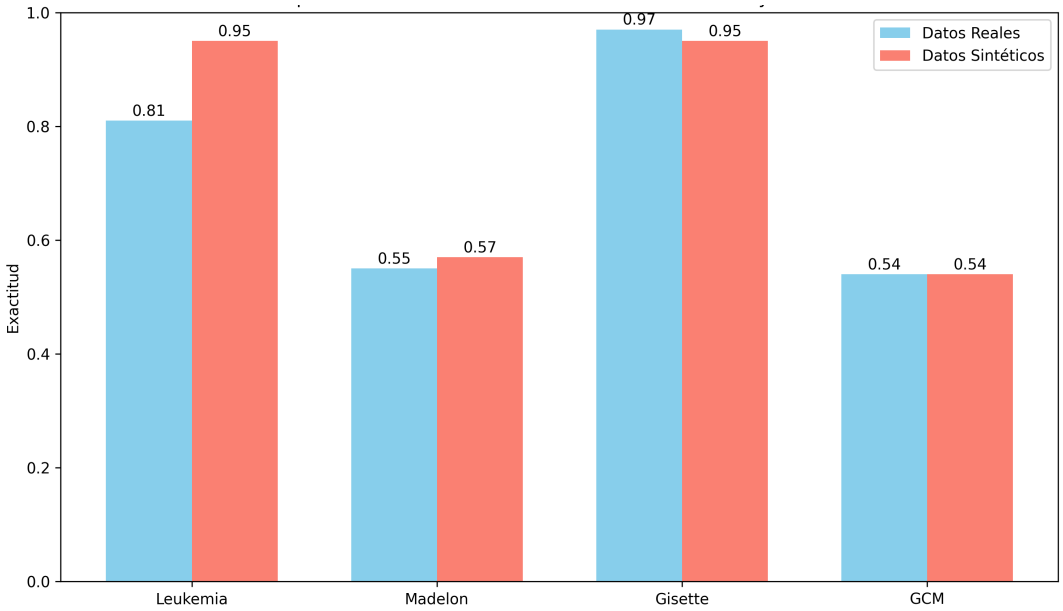


Figura 3.4 Comparación de la exactitud de MLP entrenado con datos sintéticos vs MLP entrenado con datos reales

Estos resultados son particularmente relevantes ya que sugieren que, bajo ciertas condiciones, los datos sintéticos pueden ser tan útiles como los datos reales para el entrenamiento de modelos predictivos. Como se aprecia en el gráfico, este fenómeno se observó de manera consistente en tres de los cuatro conjuntos de datos de nuestro estudio: Leukemia, Madelon y GCM, donde la precisión y la exactitud del modelo entrenado con datos sintéticos alcanzaron o superaron las métricas obtenidas con los datos originales. En el caso del dataset Gisette, el modelo entrenado con datos sintéticos estuvo muy cerca de igualar el rendimiento del modelo entrenado con datos reales.

Veamos a continuación los resultados particulares de cada conjunto de datos.

Tab.3.1 Resultados en Leukemia

| Modelo | Clase | Precisión | Recall | F1 Score | Soporte |
|---|---|---|---|---|---|
| MLP con datos reales | 0 | 0.76 | 1.00 | 0.87 | 13 |
| | 1 | 1.00 | 0.56 | 0.71 | 9 |
| Exactitud | | | | 0.81 | 22 |
| Promedio Macro | | 0.88 | 0.78 | 0.79 | 22 |
| Promedio Ponderado | | 0.86 | 0.82 | 0.80 | 22 |
| MLP con datos sintéticos | 0 | 0.93 | 1.00 | 0.96 | 13 |
| | 1 | 1.00 | 0.89 | 0.94 | 9 |
| Exactitud | | | | 0.95 | 22 |
| Promedio Macro | | 0.96 | 0.94 | 0.95 | 22 |
| Promedio Ponderado | | 0.96 | 0.95 | 0.95 | 22 |

La Tabla 3.1 presenta los resultados de las métricas de evaluación obtenidas al comparar el rendimiento de dos MLPs: uno entrenado con datos reales y otro con datos

sintéticos. Para cada modelo se muestran las métricas por clase (0 y 1) y los promedios generales. El soporte indica la cantidad de muestras por clase en el conjunto de prueba.

En cuanto a los resultados por clase, se observan diferencias significativas. Para la clase 0, el modelo entrenado con datos sintéticos logra una precisión de 0.93 frente a 0.76 del modelo con datos reales, un incremento de 17 puntos que indica una mejora sustancial en la capacidad de identificar correctamente esta clase. El recall se mantiene en 1.00 para ambos modelos, mostrando que ambos son igualmente efectivos en detectar todos los casos positivos de esta clase.

Para la clase 1, mientras que ambos modelos alcanzan una precisión perfecta de 1.00, el recall mejora notablemente en el modelo con datos sintéticos, pasando de 0.56 a 0.89. Esto significa que el modelo entrenado con datos sintéticos es significativamente mejor identificando los casos de la clase 1, reduciendo los falsos negativos.

En términos generales, estas mejoras se reflejan en un incremento de la exactitud global del modelo, que pasa de 0.81 a 0.95, y en mejoras consistentes en los promedios macro y ponderado de todas las métricas. Estos resultados sugieren que el uso de datos sintéticos no solo mejora el rendimiento general del modelo, sino que también contribuye a un aprendizaje más equilibrado entre las clases.

Tab.3.2 Resultados en Madelon

| Modelo | Clase | Precisión | Recall | F1 Score | Soporte |
|---|---|---|---|---|---|
| MLP con datos reales | 0 | 0.56 | 0.55 | 0.55 | 396 |
| | 1 | 0.54 | 0.55 | 0.55 | 384 |
| Exactitud | | | | 0.55 | 780 |
| Promedio Macro | | 0.55 | 0.55 | 0.55 | 780 |
| Promedio Ponderado | | 0.55 | 0.55 | 0.55 | 780 |
| MLP con datos sintéticos | 0 | 0.56 | 0.72 | 0.63 | 396 |
| | 1 | 0.59 | 0.42 | 0.49 | 384 |
| Exactitud | | | | 0.57 | 780 |
| Promedio Macro | | 0.58 | 0.57 | 0.56 | 780 |
| Promedio Ponderado | | 0.58 | 0.57 | 0.56 | 780 |

La Tabla 3.2 muestra los resultados comparativos entre los dos MLPs para el conjunto de datos Madelon, que presenta un caso de clasificación binaria balanceada con 396 muestras de la clase 0 y 384 de la clase 1 en el conjunto de prueba.

El análisis por clase revela patrones interesantes. Para la clase 0, mientras que la precisión se mantiene prácticamente igual (0.56 en ambos casos), el recall mejora significativamente con datos sintéticos, pasando de 0.55 a 0.72. Esto indica que el modelo entrenado con datos sintéticos es más efectivo en identificar correctamente las muestras de la clase 0.

Sin embargo, para la clase 1 observamos un comportamiento diferente: si bien la precisión mejora ligeramente de 0.54 a 0.59, el recall disminuye considerablemente de 0.55 a 0.42. Esto sugiere que el modelo con datos sintéticos es más selectivo pero menos sensible para la clase 1, lo que podría indicar un sesgo en la generación de datos sintéticos para esta clase.

A pesar de este comportamiento dispar entre clases, el modelo entrenado con datos sintéticos logra una ligera mejora en la exactitud global (de 0.55 a 0.57) y en los promedios macro y ponderado de todas las métricas. Estas mejoras, aunque modestas, son relevantes considerando que Madelon es un conjunto de datos diseñado específicamente para ser difícil de clasificar.

La Tabla 3.3 presenta los resultados para el conjunto de datos Gisette, un problema de clasificación binaria con un conjunto de prueba balanceado de 1800 muestras (904 de clase 0 y 896 de clase 1). Este dataset destaca por el alto rendimiento alcanzado por ambos modelos.

Tab.3.3 Resultados en Gisette

| Modelo | Clase | Precisión | Recall | F1 Score | Soporte |
|---|---|---|---|---|---|
| MLP con datos reales | 0 | 0.98 | 0.98 | 0.98 | 904 |
| | 1 | 0.98 | 0.98 | 0.98 | 896 |
| Exactitud | | | | 0.97 | 1800 |
| Promedio Macro | | 0.98 | 0.98 | 0.98 | 1800 |
| Promedio Ponderado | | 0.98 | 0.98 | 0.98 | 1800 |
| MLP con datos sintéticos | 0 | 0.95 | 0.97 | 0.96 | 904 |
| | 1 | 0.97 | 0.95 | 0.96 | 896 |
| Exactitud | | | | 0.95 | 1800 |
| Promedio Macro | | 0.96 | 0.96 | 0.96 | 1800 |
| Promedio Ponderado | | 0.96 | 0.96 | 0.96 | 1800 |

El modelo entrenado con datos reales muestra un rendimiento excepcional y consistente, alcanzando una precisión y recall de 0.98 para ambas clases. Esto resulta en un F1 Score de 0.98 y una exactitud global del 97%, indicando un comportamiento casi perfecto en la clasificación.

Por su parte, el modelo entrenado con datos sintéticos logra también resultados notables, aunque ligeramente inferiores. Para la clase 0, obtiene una precisión de 0.95 y un recall de 0.97, mientras que para la clase 1 estos valores se invierten (precisión 0.97, recall 0.95). Esta simetría en las métricas sugiere un comportamiento equilibrado del modelo entre ambas clases.

La diferencia de rendimiento entre ambos modelos es pequeña (2 puntos porcentuales en exactitud global) y los promedios macro y ponderado se mantienen consistentemente altos (0.96) para el modelo con datos sintéticos. Estos resultados son particularmente relevantes considerando la alta dimensionalidad del conjunto de datos Gisette, y demuestran la capacidad del AV para generar datos sintéticos de calidad incluso en espacios de características complejos.

La Tabla 3.4 presenta los resultados para GCM, un conjunto de datos particularmente desafiante que comprende 14 clases con un marcado desbalance, evidenciado en el soporte que varía desde apenas 1 muestra (clase 1) hasta 8 muestras (clase 4) en el conjunto de prueba.

El análisis por grupos de clases revela patrones distintivos:

1. Clases con mejora significativa:

- La clase 4 mantiene un rendimiento sobresaliente, mejorando de un F1 Score de 0.94 a 1.00
- La clase 6 muestra una mejora notable, pasando de un F1 Score de 0.80 a 0.89
- La clase 12 duplica su F1 Score de 0.40 a 0.86

Tab.3.4 Resultados en GCM

| Modelo | Clase | Precisión | Recall | F1 Score | Soporte |
|---|---|---|---|---|---|
| MLP con datos reales | 0 | 0.14 | 0.25 | 0.18 | 4 |
| | 1 | 0.00 | 0.00 | 0.00 | 1 |
| | 2 | 1.00 | 1.00 | 1.00 | 3 |
| | 3 | 1.00 | 0.33 | 0.50 | 6 |
| | 4 | 0.89 | 1.00 | 0.94 | 8 |
| | 5 | 0.40 | 0.67 | 0.50 | 3 |
| | 6 | 0.80 | 0.80 | 0.80 | 5 |
| | 7 | 0.50 | 0.75 | 0.60 | 4 |
| | 8 | 0.33 | 0.25 | 0.29 | 4 |
| | 9 | 0.25 | 0.67 | 0.36 | 3 |
| | 10 | 1.00 | 0.25 | 0.40 | 4 |
| | 11 | 1.00 | 0.67 | 0.80 | 3 |
| | 12 | 1.00 | 0.25 | 0.40 | 4 |
| | 13 | 1.00 | 0.20 | 0.33 | 5 |
| Exactitud | | | | 0.54 | 57 |
| Promedio Macro | | 0.67 | 0.51 | 0.51 | 57 |
| Promedio Ponderado | | 0.74 | 0.54 | 0.56 | 57 |
| MLP con datos sintéticos | 0 | 1.00 | 0.25 | 0.40 | 4 |
| | 1 | 0.00 | 0.00 | 0.00 | 1 |
| | 2 | 1.00 | 0.67 | 0.80 | 3 |
| | 3 | 1.00 | 0.17 | 0.29 | 6 |
| | 4 | 1.00 | 1.00 | 1.00 | 8 |
| | 5 | 0.43 | 1.00 | 0.60 | 3 |
| | 6 | 1.00 | 0.80 | 0.89 | 5 |
| | 7 | 0.10 | 0.25 | 0.14 | 4 |
| | 8 | 0.67 | 0.50 | 0.57 | 4 |
| | 9 | 0.50 | 0.33 | 0.40 | 3 |
| | 10 | 0.00 | 0.00 | 0.00 | 4 |
| | 11 | 1.00 | 0.67 | 0.80 | 3 |
| | 12 | 1.00 | 0.75 | 0.86 | 4 |
| | 13 | 0.21 | 0.60 | 0.32 | 5 |
| Exactitud | | | | 0.54 | 57 |
| Promedio Macro | | 0.64 | 0.50 | 0.50 | 57 |
| Promedio Ponderado | | 0.70 | 0.54 | 0.55 | 57 |

2. Clases con deterioro notable:
   - La clase 7 sufre una degradación importante, cayendo de un F1 Score de 0.60 a 0.14
   - La clase 3 muestra un deterioro significativo, reduciendo su F1 Score de 0.50 a 0.29
3. Casos especiales:

- La clase 1, con solo una muestra, mantiene un rendimiento nulo (F1 Score = 0) en ambos modelos
- La clase 0 muestra un comportamiento mixto: aumenta su precisión a 1.00 pero mantiene un recall bajo (0.25)

Un aspecto llamativo es el aumento general en la precisión con datos sintéticos: varias clases alcanzan precisión perfecta (1.00). Este patrón sugiere que el modelo con datos sintéticos es más conservador en sus predicciones, prefiriendo la precisión sobre la sensibilidad.

A pesar de estas variaciones, ambos modelos alcanzan la misma exactitud global (0.54), aunque con diferentes patrones de fortalezas y debilidades. Los promedios macro y ponderado son similares, con ligeras diferencias que reflejan el comportamiento heterogéneo entre clases.

Para concluir, es evidente que la complejidad inherente a trabajar con datos altamente desbalanceados y multiclase plantea desafíos significativos, pero incluso en estos casos los resultados sugieren que la generación sintética de datos puede mejorar el desempeño del modelo, contribuyendo a su precisión.

## 3.9. Otras consideraciones emergentes de los experimentos

Respecto de los hallazgos realizados a lo largo de todo el proceso de experimentación encontramos algunos aspectos dignos de mención.

Las pruebas con diferentes arquitecturas proporcionaron información valiosa. Como mencionamos antes, se exploraron modelos AV de tres y cuatro capas, así como AVC con múltiples capas, pero no se observaron mejoras significativas al aumentar el número de los parámetros aprendibles de los modelos. En particular, se encontró que las configuraciones más simples (e.j. 3 capas), ofrecían resultados tan buenos o incluso mejores que sus contrapartes más complejas. Esta observación refuerza la idea de que, en algunos casos, la simplicidad puede ser preferible y que el sobredimensionamiento de la arquitectura no necesariamente se traduce en mejores resultados.

Un aspecto crítico de nuestros experimentos se relaciona con la dimensionalidad del espacio latente en los modelos generativos estudiados. Durante la exploración de hiperparámetros, se observó que la configuración óptima de la variable latente no se corresponde necesariamente con espacios de mayor dimensión. Contrario a la hipótesis generalmente aceptada —según la cual "cuanto mayor la dimensión del espacio latente, mejor" (Boom et al. 2021)— nuestros hallazgos indican que, en ciertos escenarios, una representación latente más compacta puede conducir a un rendimiento superior. En concreto, los experimentos con nuestro modelo AVC demostraron que una dimensión latente de 35 unidades superaba a configuraciones con más de 50, hallazgo que se alinea además con las prácticas reportadas en (Ai et al. 2023). Este resultado, aunque aparentemente contraintuitivo, evidencia que una dimensionalidad excesiva puede introducir ruido y redundancia, comprometiendo la capacidad de generalización del modelo.

Otra observación relevante asociada con la generación sintética de datos se vincula a la cantidad optima de muestras sintéticas. En relación a esto, se constataron resultados positivos en GCM incrementando las observaciones del conjunto de datos de entrenamiento de 190 a 3000 muestras balanceadas, con 214 observaciones por clase. Esta estrategia de aumentación, donde se igualan las cantidades de observaciones de

todas las clases, es usada en Fajardo et al. (2021) con buenos resultados. En nuestros experimentos el aumento resultó en una mejora significativa en la performance del modelo, logrando igualar los resultados obtenidos con el clasificador MLP entrenado con datos reales. Sin embargo, al continuar incrementando la cantidad de datos sintéticos a 6000 muestras, se observó una degradación en el rendimiento.

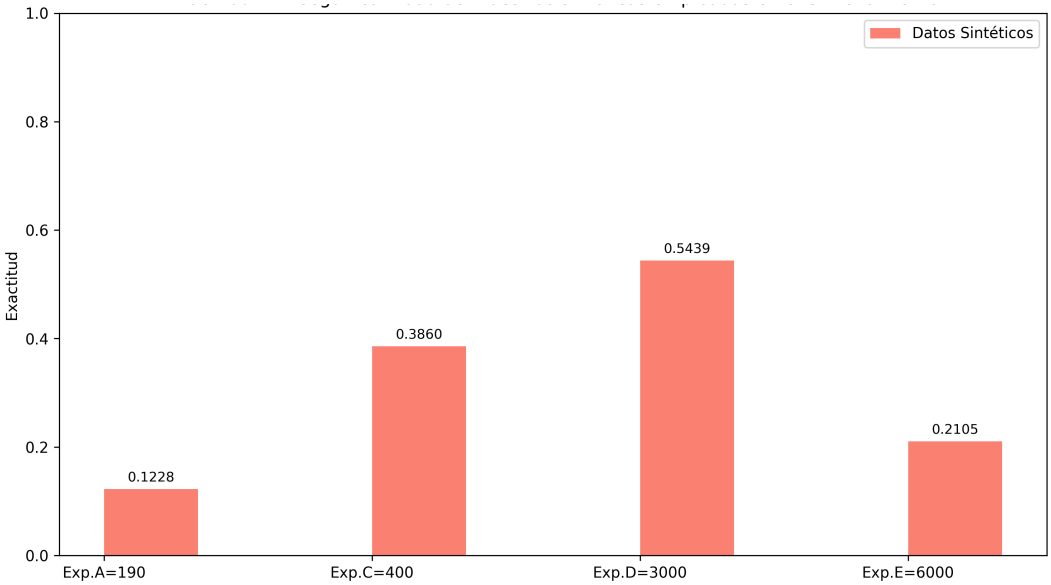


Figura 3.5 Exactitud MLP según cantidad de muestras sintéticas empleadas en el entrenamiento. Se muestran 4 diferentes experimentos con 190, 400, 3000 y 6000 muestras sintéticas.

Esto sugiere la existencia de un umbral en la cantidad de datos sintéticos que, una vez superado, empieza a producirse un sobreajuste. Este sobreajuste aumenta el error y disminuye la precisión del modelo.

Finalmente, otros emergentes que podemos destacar son los siguientes:

- la implementación de un término de paciencia tuvo un impacto positivo en la calidad de los modelos, particularmente en el caso del AVC,
- para el caso del AVC, el uso de L1_loss no proporciona un beneficio claro sobre el MSE para la tarea de generación de datos sintéticos, y
- el dropout no aporta beneficios en configuraciones ya optimizadas del AVC para este tipo de tareas.

Estos resultados llevaron a una reflexión sobre la falta de impacto positivo de ciertos ajustes, como la introducción de L1_loss y dropout. Es probable que la estabilidad y el buen rendimiento de las configuraciones ya validadas de AV y AVC, alcanzados a través de numerosos experimentos, limiten el potencial de mejora adicional mediante estos métodos.

## 3.10. Conclusiones

En primer lugar pudimos determinar que los datos sintéticos generados por un AV/AVC pueden replicar la distribución de los datos reales en los conjuntos de datos

estudiados, y por lo tanto, pueden ser tan útiles como aquellos para el entrenamiento de modelos predictivos.

Por otro lado, la búsqueda de hiperparámetros y el ajuste de la arquitectura del AV y AVC revelaron la importancia de un enfoque balanceado que evite tanto la simplicidad excesiva como la complejidad innecesaria. Los resultados obtenidos muestran que, bajo ciertas configuraciones, los datos sintéticos pueden igualar o superar la utilidad de los datos reales en la formación de modelos predictivos, aunque la eficiencia y la calidad de estos resultados dependen en gran medida de la cuidadosa calibración de los hiperparámetros y de la adecuada elección de la arquitectura del modelo.

Asimismo, los experimentos parecen reflejar que los beneficios de la aumentación de datos tienen un límite. Superado este umbral, la generación adicional de datos no solo deja de ser útil, sino que puede ser perjudicial, como se evidenció en nuestros experimentos. Este fenómeno destaca la importancia de una cuidadosa calibración en la cantidad de datos sintéticos generados, especialmente en conjuntos de datos con características complejas y de grandes dimensiones.

Capítulo 4

# Algoritmos genéticos, datos sintéticos y selección de características

En este capítulo presentamos una descripción general de los algoritmos genéticos (AG) como estrategia de selección de características, exponemos sus componentes y describimos la implementación realizada en nuestro trabajo. Seguidamente planteamos la integración del AG con la generación sintética de datos mediante autocodificaciones variacionales (AV) como estrategia de aumentación. Según el planteo hecho desde el inicio, la integración de ambas técnicas para favorecer la selección de características mediante la incorporación de datos sintéticos permitiría resolver problemas de alta dimensionalidad, escasez muestral y ruido. Para terminar, compartimos los experimentos realizados sobre esta integración entre AV y AG, y presentamos los resultados obtenidos a fin de determinar hasta qué punto hemos podido resolver los problemas motivantes de este trabajo.

## 4.1. Elementos básicos de los algoritmos genéticos

Los AG son una clase de algoritmo inspirado en la evolución biológica y en la teoría de la selección natural. Los mismos se basan en el concepto de evolución de una población de soluciones potenciales a lo largo de múltiples generaciones. Utilizando operadores genéticos como la selección, el cruce y la mutación los AG generan nuevas soluciones a partir de las disponibles, mejorando su calidad y favoreciendo la preeminencia de ciertas características. Una breve descripción del algoritmo se presenta en el diagrama que se muestra a continuación.

En la Figura 4.1 se ilustra la estructura general de un algoritmo genético. En términos generales, se parte de un conjunto de datos (por ejemplo, un vector $(x)$) y se crea una población inicial de soluciones candidatas (individuos), cada una codificada en un cromosoma (por ejemplo, una cadena binaria). A continuación, cada individuo se evalúa mediante una función de aptitud (fitness) que refleja la calidad de la solución propuesta. Si no se cumple la condición de terminación (que puede estar representada por un umbral de error o un número máximo de iteraciones), se aplican las fases de selección, cruce y mutación para generar una nueva generación de individuos potencialmente más aptos. Estos pasos se repiten de manera iterativa hasta que se cumpla el criterio de terminación. Para ilustrar lo anterior, consideremos la función $y = -x^2$ y el objetivo de maximizar su valor. En este caso, cada individuo del algoritmo genético representa un posible valor de $x$ codificado en un cromosoma binario. La aptitud de cada individuo se calcula como $y = -x^2$.

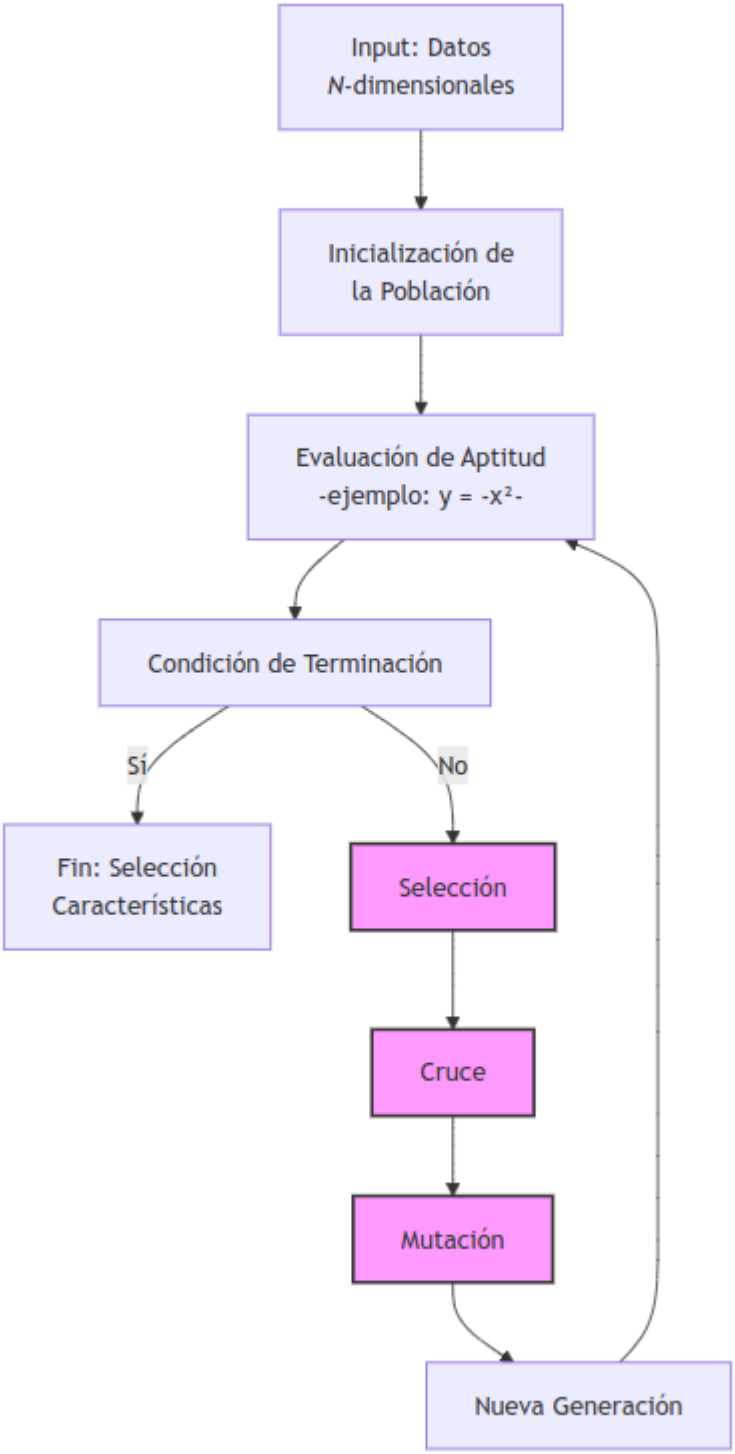


Figura 4.1. Diagrama de un Algoritmo Genético

Tras las etapas de selección, cruce y mutación, los valores de $x$ que producen valores más altos de $-x^2$ tienden a sobrevivir y propagarse. Al cabo de varias iteraciones, el algoritmo converge hacia el valor $x = 0$, que es la solución que maximiza $-x^2$. De este modo, el diagrama de la figura refleja cómo, a través de procesos inspirados en la evolución biológica, se van refinando las soluciones hasta cumplir cierto objetivo.

A pesar de su extraordinaria simpleza -o quizás gracias a ella-, los AG constituyen algoritmos robustos, capaces de encontrar soluciones efectivas en una amplia variedad de problemas de optimización. Dicha robustez está determinada, como bien sostiene Goldberg (1989), por una serie de características distintivas, que fortalecen su configuración de búsqueda, a saber: a) operan sobre un espacio codificado del problema y no sobre el espacio en su representación original; b) realizan la exploración evaluando una población de soluciones y no soluciones individuales; c) tienen como guía una función objetivo (también llamada función de aptitud) que no requiere derivación u otras funciones de cálculo; y d) suponen métodos probabilísticos de transición (operadores estocásticos) y no reglas determinísticas. Estas características permiten a los AG superar restricciones que tienen otros métodos de optimización, condicionados -por ejemplo- a espacios de búsqueda continuos, diferenciables o unimodales. Por ello, su aplicación se ha difundido notablemente, trascendiendo los problemas clásicos de optimización, aplicándose en distintas tareas (Vie, Kleinnijenhuis, and Farmer 2021) y a lo largo de diversas industrias (Jiao et al. 2023).

Para comprender la eficacia que tienen los AG como método de búsqueda, veamos en detalle los puntos mencionados anteriormente.

## 4.2. Codificación del espacio de soluciones

Como señalamos, los AG se distinguen de otros algoritmos por su capacidad para operar en un espacio codificado del problema, en lugar de operar directamente sobre el espacio en su representación original. Esto sucede gracias a la transformación de las soluciones potenciales en cadenas binarias, comúnmente conocidas como cromosomas, que luego son objeto de transformación mediante operadores genéticos como la mutación y el cruce. La capacidad de los AG para operar con estas representaciones codificadas determina su adaptabilidad y eficacia en una amplia gama de problemas de optimización.

La codificación adecuada del problema es un paso inicial clave para el correcto desempeño del algoritmo. La elección de la codificación depende de la naturaleza del problema y de las características de las soluciones que se buscan optimizar. Por ejemplo, en el tratamiento de información genética una opción frecuente es el uso de codificación binaria. En esta representación, cada cromosoma es una cadena de bits (0s y 1s) donde cada posición o bit representa la presencia (1) o ausencia (0) de una característica particular en la solución. Así, en un problema de selección de características con 100 variables, cada cromosoma sería una cadena de 100 bits donde un 1 indica que esa característica es seleccionada y un 0 que no lo es. Esta codificación binaria es particularmente eficiente para problemas de selección ya que permite representar de manera natural el espacio de búsqueda como un plano n-dimensional, donde cada dimensión representa una posible combinación de características. Además, los operadores genéticos como el cruce y la mutación pueden implementarse de manera directa y eficiente sobre estas cadenas binarias.

Dada la importancia que tiene la codificación, es fácil advertir que así como una elección adecuada de la estrategia de codificación puede facilitar la convergencia del AG hacia buenas soluciones, una elección inadecuada puede tener consecuencias negativas en su desempeño. En efecto, una codificación inapropiada puede llevar a una exploración ineficaz del espacio de soluciones, generando soluciones redundantes o incluso inviables. Una codificación que no preserve la viabilidad de las soluciones durante

la evolución, puede resultar en la convergencia prematura del AG hacia soluciones subóptimas.

La traducción entre la representación interna codificada (genotipo) y la solución en el contexto del problema (fenotipo) es un componente importante del diseño de los AG. El genotipo se refiere a la representación interna de una solución en el Algoritmo Genético (AG). Es el "cromosoma" o la estructura de datos que codifica la información genética de un individuo, comúnmente representado como una cadena de bits (0s y 1s). Por otro lado, el fenotipo es la manifestación externa del genotipo en el contexto del problema, correspondiendo a la solución real en el espacio de búsqueda que se evalúa mediante la función de aptitud para determinar la calidad del individuo. Por ejemplo, en un problema de selección de características, cada bit del genotipo indica la inclusión (1) o exclusión (0) de una característica específica. Consideremos el genotipo 1100101, que representa la selección de las características 1, 2, 5 y 7, excluyendo las características 3, 4 y 6. El fenotipo asociado a este genotipo sería el subconjunto de características seleccionadas [X1, X2, X5, X7], el cual se utilizará para entrenar un modelo. El desempeño de este modelo, evaluado mediante la función de aptitud, determinará la calidad del individuo en el proceso evolutivo del AG. El mapeo descrito anteriormente no solo permite interpretar las soluciones generadas por el algoritmo, sino que también influye en la eficacia de los operadores genéticos. Ello así, debido a que los operadores genéticos actúan directamente sobre la representación codificada, lo que puede afectar la exploración del espacio de soluciones y la convergencia del AG.

Una de las principales ventajas de operar en un espacio codificado del problema radica en la posibilidad de aplicar operadores genéticos de manera eficiente, lo que permite una exploración conveniente del espacio de soluciones. En efecto, los operadores genéticos -que veremos en breve- son diseñados específicamente para actuar directamente sobre la representación codificada, generando nuevas soluciones de manera efectiva.

Un proceso típico de codificación y decodificación en un AG incluye los siguientes pasos:

1. Codificación del problema: Representación directa del problema, por ejemplo, valores continuos o categóricos [1].
2. Operadores Genéticos: Aplicación de mutación, cruce y selección en la representación codificada.
3. Decodificación: Traducción inversa de la solución codificada al espacio original para evaluación.

[1] Cabe aclarar que en el caso de espacios continuos, el AG debe establecer una resolución o granularidad específica para la representación discreta del problema. Esto se debe a que un cromosoma binario solo puede representar un número finito de valores dentro del rango establecido, no cualquier valor continuo arbitrario. La elección de esta granularidad afecta directamente la exactitud con la que el AG puede aproximar soluciones en el espacio continuo.

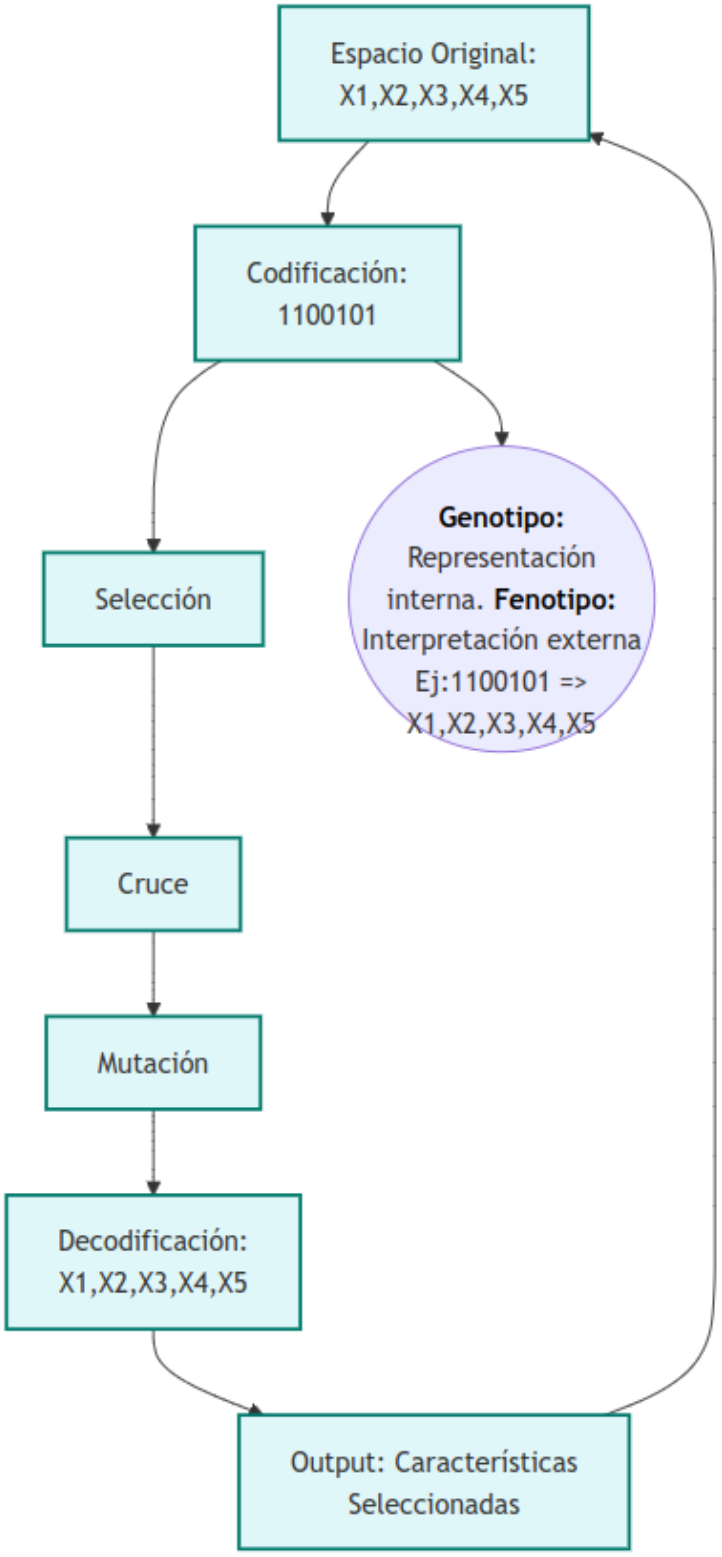


Figura 4.2. Diagrama de codificación y decodificación

En el caso de nuestra investigación, dada la alta dimensionalidad de los datos y la

complejidad de los modelos, la codificación adecuada de las soluciones fue un proceso fundamental para permitir que los AG pudieran encontrar buenas soluciones o cercanas al óptimo en tiempo razonable.

## 4.3. Búsqueda por población de soluciones

Otra característica distintiva de los AG es su enfoque en la evaluación de una población de soluciones en cada iteración, en lugar de centrarse en una única solución. Esta población de soluciones, también conocida como población de individuos, permite a los AG explorar simultáneamente múltiples regiones del espacio de búsqueda, aumentando así la probabilidad de encontrar buenas soluciones o cercanas al óptimo.

Como vimos en el ejemplo precedente, la población inicial regularmente se genera de manera aleatoria, y cada individuo dentro de esta población representa una solución potencial al problema. A lo largo de las generaciones, los AG aplican operadores genéticos como selección, cruce y mutación para producir nuevas generaciones de individuos, mejorando iterativamente la calidad de las soluciones.

La diversidad genética dentro de la población es fundamental para la eficacia de los AG, ya que permite a los algoritmos explorar de manera más exhaustiva el espacio de características y evitar la convergencia prematura hacia soluciones subóptimas. En efecto, consideremos una población homogénea donde todos los individuos son idénticos. En este caso, la capacidad del AG para explorar nuevas regiones del espacio de búsqueda se ve severamente limitada, lo que puede resultar en una convergencia temprana hacia soluciones subóptimas. Por el contrario, una población diversa, donde cada individuo representa una solución única, permite al AG explorar una variedad de soluciones y adaptarse a las condiciones cambiantes del problema.

A modo de ejemplo consideremos estas dos poblaciones de 5 individuos codificados como sigue:

Población A, con 5 individuos de longitud 5, de alta diversidad:

- Individuo 1: `11001`
- Individuo 2: `10110`
- Individuo 3: `01101`
- Individuo 4: `11100`
- Individuo 5: `00011`

Población B, con 5 individuos de longitud 5, de baja diversidad:

- Individuo 1: `11111`
- Individuo 2: `11111`
- Individuo 3: `11011`
- Individuo 4: `11010`
- Individuo 5: `11010`

Como podemos advertir en este ejemplo, cada individuo representa una solución potencial al problema, donde cada bit en la cadena codificada corresponde a una característica que puede ser seleccionada o excluida. En estas poblaciones los individuos de A son distintos entre sí, lo que permite al AG combinar cromosomas distintos y explorar nuevas zonas del espacio de búsqueda. Por el contrario, los individuos de B son idénticos, lo que limita la capacidad del AG para explorar nuevas regiones del espacio de búsqueda.

## 4.4. Función de aptitud y evaluación de soluciones

La función de aptitud es el núcleo que dirige el proceso evolutivo en los AG, determinando las probabilidades de supervivencia y reproducción de las soluciones. Su diseño y correcta implementación son esenciales para asegurar que el AG no solo converja hacia soluciones de alta calidad, sino que también lo haga de manera eficiente y efectiva, especialmente en problemas donde las evaluaciones de aptitud son costosas o complejas.

En el proceso evolutivo de los AG, la función de aptitud se aplica al fenotipo de cada solución, es decir, a su manifestación en el contexto del problema a resolver, después de que el genotipo (la representación codificada de la solución) ha sido transformado. Esta evaluación cuantifica qué tan bien una solución potencial cumple con los objetivos del problema, asignándole un valor numérico que refleja su desempeño relativo en comparación con otras soluciones dentro de la población.

El diseño de la función de aptitud es un aspecto crítico del proceso de modelado en los AG, ya que guía la dirección de la búsqueda evolutiva. Específicamente, la función de aptitud debe estar alineada con los objetivos del problema, reflejando correctamente las restricciones necesarias a satisfacer. En situaciones de optimización multiobjetivo, donde varios criterios deben ser optimizados simultáneamente, es común que funciones de aptitud individuales se combinen en una única métrica a través de técnicas como la suma ponderada de los valores de aptitud individuales. En el contexto de nuestra investigación, orientada a la selección de características, la función de aptitud combina la aptitud de un individuo en términos de exactitud y el tamaño del conjunto de características seleccionadas (veremos un ejemplo en breve).

En línea con lo anterior, la evaluación precisa de las soluciones mediante la función de aptitud puede constituir un proceso sujeto a múltiples restricciones. Aunque la asignación de valores de aptitud más bajos a soluciones de baja calidad y más altos a soluciones superiores pueda parecer un criterio ineludible, en la práctica, este proceso requiere comúnmente consideraciones adicionales. Por ejemplo, en problemas con restricciones, una solución con una aptitud alta, pero que infrinja requerimientos del problema, debe recibir una calificación de aptitud inferior a una solución menos apta pero libre de tales infracciones. De esa forma, el AG puede orientar la búsqueda hacia soluciones viables. Con esa lógica, en la optimización multiobjetivo es necesario establecer criterios para ponderar la contribución de cada objetivo, especialmente cuando los distintos objetivos compiten entre sí.

## 4.5. Operadores estocásticos y esquemas genéticos

Como hemos señalado, los AG emplean métodos probabilísticos de transición conformados por operadores estocásticos, que introducen aleatoriedad en el proceso evolutivo. Esto determina que las transformaciones dentro de un AG no siguen un camino determinista hacia la solución óptima; en su lugar, cada generación de soluciones es producto de un proceso estocástico controlado.

Los operadores genéticos fundamentales en este proceso son la selección, el cruce y la mutación. Los mismos son responsables de la generación de nuevas soluciones, e inciden directamente en la evolución de los patrones genéticos que los AG tienden a preservar y reproducir. Patrones que se conocen como esquemas (Goldberg, David E. 1989).

Según explica Goldberg, los esquemas son estructuras genéticas que se repiten en la población y que influyen en la evolución de los individuos. Estos esquemas pueden ser de orden bajo (pocos genes) o de orden alto (más genes), y de longitud de definición baja (pocos bits) o de longitud de definición alta (más bits). En su operatoria, los AG tienden a favorecer los esquemas de orden bajo y longitud de definición baja que muestran un rendimiento mejor que la media. Este fenómeno, conocido como Teorema del Esquema, proporciona una base para entender cómo la selección y los operadores estocásticos actúan en conjunto para guiar la evolución hacia buenas soluciones.

Pasando a ver cada uno de los operadores, tenemos que la selección opera identificando y preservando los esquemas con aptitudes superiores a la media de la población. En términos probabilísticos, los esquemas con mejor aptitud tienen una mayor probabilidad de ser seleccionados y reproducidos en la siguiente generación. Esta selección basada en aptitud es clave para mantener y amplificar características beneficiosas dentro de la población. Dentro de los operadores de selección encontramos diversas estrategias, entre las que se destacan: selección por ruleta, torneo y ventana:

- selección por ruleta asigna a cada individuo una probabilidad de ser seleccionado proporcional a su aptitud relativa dentro de la población. Imaginando una ruleta dividida en segmentos donde el tamaño de cada segmento corresponde a la aptitud del individuo, los individuos con mayor aptitud ocupan una mayor porción de la ruleta, aumentando sus probabilidades de ser elegidos. Este método favorece fuertemente a los individuos más aptos, promoviendo una rápida explotación de soluciones prometedoras. Sin embargo, puede llevar a una reducción de la diversidad genética si ciertos individuos dominan consistentemente el proceso de selección.
- selección por torneo consiste en seleccionar al azar un subconjunto de individuos de la población y elegir al mejor de este grupo para la reproducción. Este método introduce un equilibrio entre explotación y exploración, ya que permite que individuos con alta aptitud tengan una mayor probabilidad de ser seleccionados, mientras que también da oportunidad a individuos menos aptos de participar ocasionalmente. La presión de selección puede ajustarse variando el tamaño del torneo, donde torneos más grandes incrementan la probabilidad de seleccionar a los individuos más aptos, mientras que torneos más pequeños fomentan una mayor diversidad genética.
- selección por ventana divide la población en grupos o "ventanas" y selecciona a los individuos más aptos dentro de cada ventana para la reproducción. Este enfoque asegura una representación equitativa de diferentes regiones del espacio de búsqueda, evitando que solo los individuos de alta aptitud global dominen el proceso de selección. Al mantener la diversidad entre las ventanas, la selección por ventana promueve una exploración más amplia del espacio de soluciones, lo que puede ser especialmente beneficioso en problemas con múltiples óptimos locales.

La selección por sí sola no es suficiente para garantizar la exploración global del espacio de búsqueda, de ahí la importancia del cruce y la mutación.

El operador de cruce permite la recombinación de material genético entre dos o más soluciones. En un AG, la función principal del cruce es preservar y mejorar las características exitosas encontradas en los padres, mientras introduce suficiente variación para explorar nuevas áreas del espacio de búsqueda. Por ejemplo, en la representación binaria, un cruce de un punto dividirá dos soluciones en una posición elegida aleatoriamente y combinará segmentos de ambas para crear nuevos individuos. Este

proceso asegura la transmisión de esquemas de orden bajo y longitud de definición baja, mientras introduce nuevas combinaciones genéticas que pueden llevar a soluciones más adaptadas.

En la Figura 4.3 se muestra un ejemplo de cruce de un punto entre dos soluciones binarias:

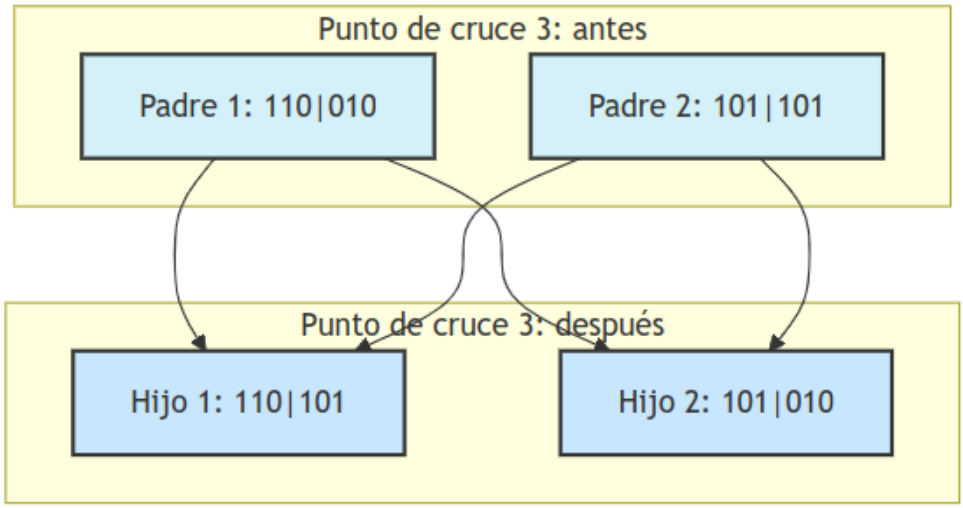


Figura 4.3 Diagrama de cruce de un punto

En este caso, el cruce de un punto en la posición 3 divide los padres en dos segmentos y combina los segmentos para generar dos nuevos individuos. Este proceso de cruce permite la recombinación de material genético entre los padres, preservando y mejorando las características exitosas encontradas en ellos.

El operador de mutación, por su parte, introduce cambios aleatorios en las soluciones existentes, actuando como un mecanismo de perturbación que permite al AG escapar de óptimos locales y explorar más exhaustivamente el espacio de soluciones. La mutación puede variar desde simples alteraciones de bits en cadenas binarias hasta ajustes en representaciones continuas mediante la adición de ruido gaussiano. La mutación es crucial para asegurar que el AG mantenga la capacidad de descubrir nuevas áreas del espacio de búsqueda.

Un ejemplo de mutación en una solución binaria sería:

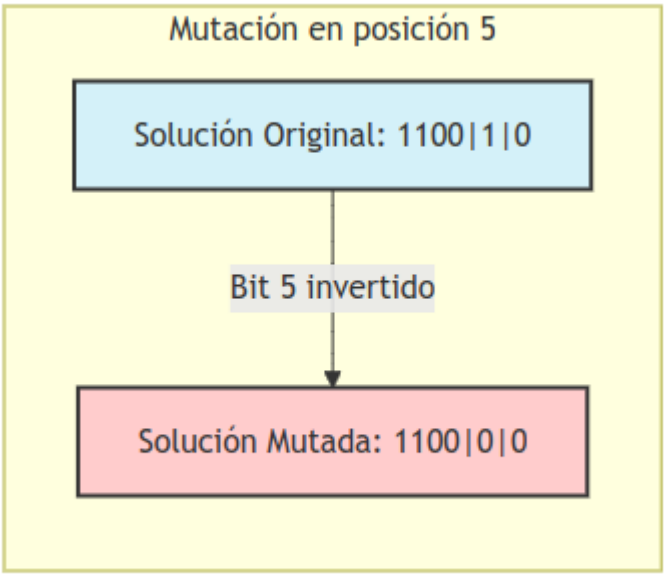


Figura 4.4 Diagrama de mutación

A esta altura ha de ser evidente que la preservación de ciertos patrones genéticos de aptitud superior es fundamental para la evolución de la población en un AG. La teoría de los esquemas, que se basa en el concepto de esquemas genéticos, proporciona un marco formal para entender cómo los operadores genéticos actúan en conjunto para guiar la evolución hacia buenas soluciones. Goldberg, David E. (1989) nos presenta, en relación a este punto, la idea del Teorema del Esquema como una herramienta teórica que permite predecir la evolución de los esquemas en una población a lo largo de múltiples generaciones. Este teorema tiene en cuenta factores como la aptitud de los esquemas, la probabilidad de cruce y mutación, la longitud de definición y el orden de los esquemas, y proporciona una guía para entender cómo los esquemas se propagan y se mantienen en la población. Para una revisión más detallada del teorema del esquema, remitimos al lector al Apéndice B.

Hasta aquí hemos expuesto los fundamentos teóricos de los AG, que nos permitirán entender la implementación de los mismos en la siguiente sección.

## 4.6. Integración de Autocodificadores Variacionales y Algoritmos Genéticos

En esta sección decribimos la integración de AV y AG como estrategia que puede favorecer la selección de características en contexto donde los datos pueden ser caracterizados por alta dimensionalidad y escasez muestral, desbalance de clases y ruido. La estrategia propuesta, representada en la Figura 4.5, incluye la generación de datos sintéticos mediante AV para aumentar el número de muestras originales y de esta forma mejorar la selección de características con AG. Estos datos aumentados contienen las observaciones originales y también las observaciones sintéticas creadas a partir de ellas.

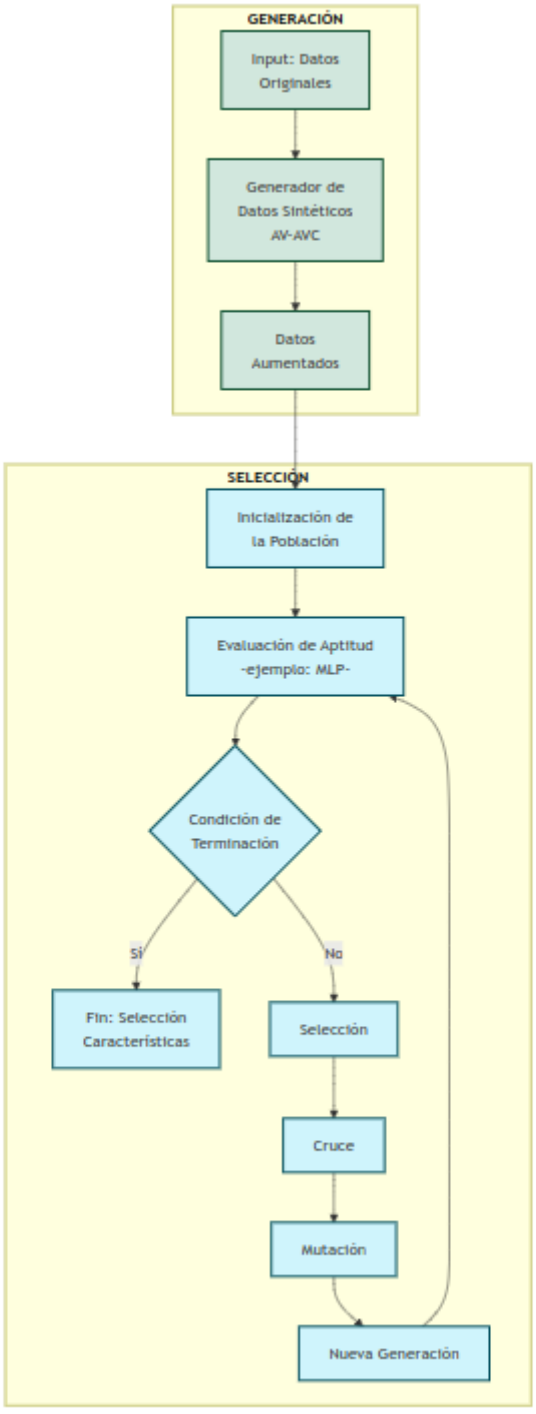


Figura 4.5 Diagrama de flujo de AV - AG

El proceso comienza con la etapa de generación de datos sintéticos donde se emplea un AV o AVC según el tipo de problema a resolver (clasificación binaria o multiclase). En esta etapa el modelo de autocodificador es entrenado con los datos originales, y parametrizado según una configuración óptima definida. En nuestro caso, dicha configuración se obtuvo mediante el proceso de optimización de hiperparámetros descrito en el capítulo anterior.

Seguidamente, se emplea el modelo de autocodificador entrenado para generar N datos

sintéticos, que se concatenan con los datos originales. Ya veremos que el valor de N no es trivial, dado su impacto en los resultados del AG. Como consecuencia de esta etapa, tenemos como nueva integración de los datos un conjunto que incluye tanto instancias originales y como instancias sintéticas. Finalmente, se emplea un AG para la selección de características empleando como entrada este dataset aumentado como se puede ver en la Figura 4.5.

Este primer paso vinculado a la generación de datos sintéticos mediante AV incluye tanto el entrenamiento del modelo como la generación. Su objetivo es aumentar el número de muestras disponibles, y de esa forma superar el problema de la escasez muestral y el desbalance de clases. Al mismo tiempo, dado que en la tarea de generación los autocodificadores involucran un proceso de codificación-decodificación siguiendo la distribución de probabilidad de los datos originales, se espera que las nuevas instancias sintéticas reduzcan el ruido presente en dichos datos.

El entrenamiento de los modelos en la etapa generativa fue un entrenamiento estándar donde se utilizaron las particiones originales de entrenamiento y testeo, subdividiendo la primera en dos subconjuntos, uno para entrenamiento y otro para validación.

Para la configuración óptima de los modelos AV y AVC, se implementó una estrategia de BO, tal como se detalló en el capítulo anterior. Esta metodología fue seleccionada por su eficiencia para explorar espacios de hiperparámetros amplios, logrando un equilibrio entre exploración y explotación que resultó particularmente ventajoso dado el tamaño del espacio de búsqueda. Para el modelo AV, se exploraron rangos entre de entre 50 y 5000 neuronas para las capas 2 y 3, mientras que para la capa latente se consideraron dimensiones entre 10 y 300. En el caso del modelo AVC, los rangos explorados fueron de 50 a 3000 neuronas para las capas 2 y 3, y de 10 a 1000 para la dimensión de la capa latente. La tasa de aprendizaje se exploró en un rango entre $10^{-}5$ y $10^{-}2$, permitiendo cubrir órdenes de magnitud que típicamente resultan efectivos en el entrenamiento de redes neuronales profundas. Esta estrategia de búsqueda sistemática nos permitió identificar configuraciones óptimas para cada conjunto de datos, maximizando tanto la capacidad generativa como la calidad de las representaciones latentes.

En la Tabla 4.1 se presentan los parámetros de los modelos de AV y AVC configurados para cada conjunto de datos:

| dataset | modelo | capa | neuronas | tasa_aprend. | epocas |
|---|---|---|---|---|---|
| Leukemia | AV | 1 | 7129 | 0.00026927 | 2889 |
| | | 2 | 346 | | |
| | | 3 | 178 | | |
| | | latente | 108 | | |
| Madelon | AV | 1 | 500 | 0.00015503 | 1364 |
| | | 2 | 835 | | |
| | | 3 | 308 | | |
| | | latente | 25 | | |
| Gisette | AV | 1 | 5000 | 0.00094609 | 2739 |
| | | 2 | 3870 | | |
| | | 3 | 2987 | | |
| | | latente | 18 | | |
| GCM | AVC | 1 | 16063 | 0.00068850 | 3613 |
| | | 2 | 358 | | |

| dataset | modelo | capa | neuronas | tasa_aprend. | epocas |
|---|---|---|---|---|---|
| | | 3 | 189 | | |
| | | latente | 35 | | |
| ALL | AVC | 1 | 12600 | 0.00040148 | 949 |
| | | 2 | 434 | | |
| | | 3 | 176 | | |
| | | latente | 35 | | |

Tabla 4.1 Parámetros de los modelos de AV y AVC

Las mismas dimensiones se emplearon para las capas de las redes codificadora y decodificadora que componen los modelos de AV y AVC.

En cuanto al preprocesamiento de los datos, se aplicó una normalización estándar (mediante `StandardScaler`, scikit-learn) durante entrenamiento y validación a cada conjunto para homogenizar las escalas de las características y favorecer la convergencia del modelo. Esta normalización consiste en sustraer la media y dividir por la desviación estándar de cada variable, minimizando así el efecto de distintas magnitudes o unidades de medida.

Avanzando al segundo paso, la selección de características se realiza utilizando un AG con datos aumentados. La implementación se centra en la optimización simultánea de la exactitud de un modelo MLP y la reducción del número de características seleccionadas, utilizando los operadores genéticos vistos en la sección anterior.

La función de aptitud empleada en nuestro algoritmo genético está diseñada para optimizar simultáneamente dos objetivos: maximizar la exactitud del modelo de clasificación y minimizar el número de características seleccionadas. Esta función se define como una combinación ponderada de estos dos objetivos:

$$f = \alpha \cdot \text{acc} + (1 - \alpha) \cdot \text{n\_genes}, \tag{4.1}$$

donde:
- acc representa la exactitud del modelo de clasificación (exactitud para clasificación binaria o exactitud balanceada para clasificación multiclase)
- n_genes representa la proporción de características no seleccionadas (calculada como $1 - \frac{\text{características seleccionadas}}{\text{total de características}}$)
- $\alpha$ es un parámetro de ponderación que controla el equilibrio entre exactitud y reducción del número de características.

En todos nuestros experimentos, el parámetro de ponderación $\alpha$ se estableció en 0.5, asignando igual importancia a ambos componentes: la exactitud del modelo y la reducción del número de características. Esta configuración refleja nuestro objetivo de encontrar un subconjunto de características que no solo maximice el rendimiento predictivo, sino que también simplifique el modelo resultante.

La configuración del AG a lo largo de los distintos experimentos que formaron parte de esta investigación (y que revisaremos en detalle en el próximo punto) incluyó:

- Población inicial: individuos generados aleatoriamente, cada uno representado como una lista de bits de longitud igual al número de características.
- Función de aptitud: Maximización de la exactitud del modelo de clasificación y minimización del número de características activas.

- Operadores genéticos: Selección por torneo, cruce de dos puntos y mutación por inversión de bits.
- Parámetros del AG: Probabilidad de mutación (e.g.`PROB_MUT = 0.1`), probabilidad de cruce (e.g. `PX = 0.75`), número máximo de generaciones (e.g.`GMAX = 15`).
- Evaluación de características: Análisis de la frecuencia de activación de las características a lo largo de las generaciones.
- Criterio de terminación: Convergencia o número máximo de generaciones alcanzado.
- Análisis de resultados: Selección de las características más relevantes basadas en su frecuencia de activación.

En cada experimento, la implementación del AG comienza con la definición de los componentes básicos del algoritmo. Se define una función de aptitud (`fitness`) orientada a maximizar, que evalúa cada individuo en función de dos criterios: la exactitud del modelo de clasificación entrenado con las características seleccionadas, y la fracción de características activas. Esta función de aptitud está diseñada para balancear la necesidad de un modelo predictivo preciso con la simplicidad y la eficiencia del modelo, evitando el sobreajuste y facilitando la interpretación del modelo final.

Los individuos, representados como listas de bits, se construyen utilizando una función de construcción de genes que genera un bit aleatorio basado en una probabilidad definida (`p_indpb`). Estos individuos se agrupan en una población inicial, que luego se somete a un proceso evolutivo. Durante la evolución, los individuos se seleccionan mediante la técnica de torneo, donde aquellos con mejor aptitud tienen una mayor probabilidad de ser elegidos para reproducción. Los individuos seleccionados se cruzan utilizando un operador de cruce de dos puntos (`cxTwoPoint`), que intercambia segmentos de los cromosomas de los padres para generar descendientes con combinaciones genéticas novedosas. Posteriormente, se aplica un operador de mutación que invierte los bits en el cromosoma según la probabilidad de mutación definida, asegurando que el AG mantenga la capacidad de explorar nuevas regiones del espacio de búsqueda.

A lo largo de las generaciones, el AG monitoriza y registra diversas estadísticas de la población, como la aptitud promedio, la exactitud y el número de genes activos. Estas métricas permiten evaluar el progreso del algoritmo y la convergencia hacia buenas soluciones. Al final del proceso evolutivo, se realiza un análisis de las características seleccionadas, calculando la frecuencia de activación de cada característica a lo largo de las generaciones y seleccionando las más recurrentes como las más relevantes. Este enfoque permite identificar un subconjunto óptimo de características que no solo maximiza la exactitud del modelo, sino que también minimiza su complejidad.

Con fines ilustrativos, en el Apéndice A se presenta un script genérico de un Algoritmo Genético implementado con la librería `DEAP` de Python, que puede ser adaptado para la selección de características en problemas de alta dimensionalidad.

## 4.7. Experimentos realizados y sus resultados

A continuación presentamos los experimentos realizados en el marco de la investigación, cuyo objetivo principal fue evaluar la efectividad de la técnica de aumentación de datos en la selección de características mediante AG en contextos de alta dimensionalidad y escasez muestral, desbalance de clases y ruido. Para ello, se diseñaron

y ejecutaron experimentos utilizando cuatro conjuntos de datos distintos: Leukemia, Gisette, Madelon y GCM (descriptos en el Capítulo 2), cada uno representando diferentes desafíos en términos de tamaño y características de los datos. Un quinto conjunto de datos, ALL-Leukemia, fue utilizado para validar los resultados obtenidos en el contexto de clasificación multiclase.

El enfoque experimental adoptado fue comparativo, contrastando el desempeño de los AG en la selección de características utilizando datos originales frente a la misma tarea utilizando datos aumentados con muestras sintéticas generados por un AV o AVC según el tipo de problema a resolver (clasificación binaria o multiclase). Los parámetros de los AG fueron mantenidos constantes entre los grupos de experimentos con y sin aumento, permitiendo una evaluación directa del impacto de la aumentación.

## 4.8. Metodología seguida en los experimentos

Los experimentos se diseñaron para comparar el rendimiento de los AG aplicados a los datos originales y a un conjunto de datos aumentado con muestras sintéticas. El objetivo de esta comparación era evaluar el impacto de la aumentación de datos en el proceso de selección de características. Para ello, se utilizaron los siguientes criterios de evaluación: exactitud en la clasificación, número de características seleccionadas y estabilidad de la selección de características. Con el fin de asegurar la comparabilidad de los resultados, siempre se empleó la partición original de testeo para evaluar la exactitud del algoritmo.

Se llevaron a cabo experimentos utilizando el modelo de AG descrito previamente, adoptando una representación binaria para las características. Para encontrar la configuración más apropiada del AG para cada conjunto de datos se realizaron experimentos exploratorios a partir de los cuales se obtuvieron los parámetros del algoritmo. La función de aptitud se basó en un clasificador de perceptrón multicapa (MLP), que evaluaba la exactitud de la clasificación y penalizaba el número de características seleccionadas, ambos componentes con igual ponderación ( $= 0.5$). El objetivo del AG era identificar un subconjunto óptimo de características que maximizara la exactitud y minimizara la dimensionalidad del espacio de características.

El aspecto innovador de este enfoque está en el uso de un AV para enriquecer el proceso evolutivo del AG. El AV, según vimos en el Capítulo 3, fue optimizado para generar muestras sintéticas que preservaran la estructura subyacente de los datos originales, y permitiera al AG explorar un espacio aumentado de características. La hipóstesis que se buscó validar en el presente trabajo fue que: aumentar sintéticamente el número de muestras disponibles permitiría al AG identificar un subconjunto más relevante de características, que se traduciría en una mejora en la exactitud y la eficiencia de la selección.

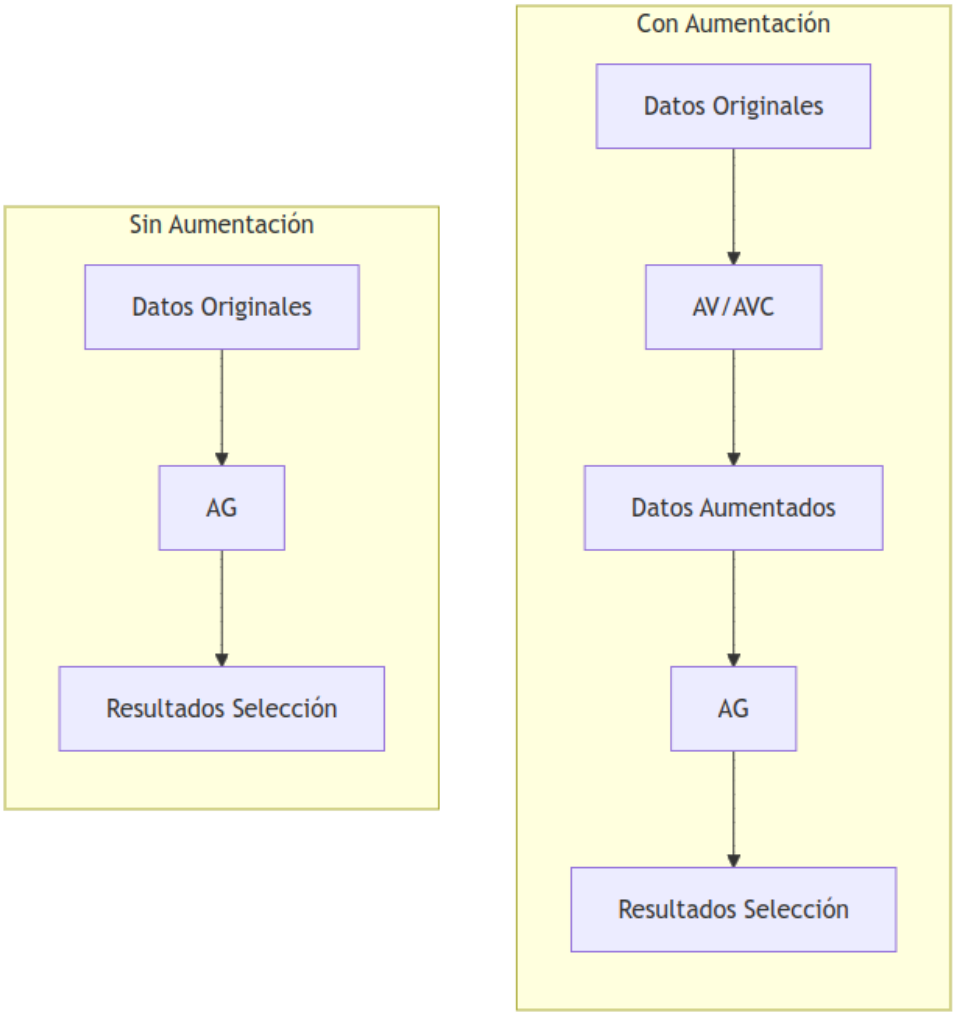


Figura 4.6. Comparación de experimentos

Para comunicar los resultados de nuestra investigación hemos elegido el gráfico de cajas para representar los resultados y permitir la comparación entre experimentos realizados con datos originales y aquellos donde se utilizaron datos aumentados. Esta elección ofrece varias ventajas: facilita la visualización de la distribución completa de los resultados, permite identificar claramente la mediana y los cuartiles de cada grupo experimental, y proporciona una representación visual de la variabilidad en los resultados, ayudando a evaluar la estabilidad de las soluciones encontradas. Así, en las figuras de la presente sección cada punto en el gráfico de cajas representa un experimento e indica la exactitud promedio lograda y cantidad promedio de características seleccionadas en la población al finalizar el proceso evolutivo.

## 4.9. Leukemia

Según lo visto en el Capítulo 1, el conjunto de datos Leukemia de expresión génica obtenidos de micro-datos de ADN es reconocido por su alta dimensionalidad (7129 mediciones) en relación al número de muestras (38 para entrenamiento y 34 para testeo). Esto lo convierte en un candidato apropiado para evaluar la capacidad de los AV

para generar datos sintéticos, a partir de los cuales aumentar el número de muestras disponibles y mejorar el desempeño de los AG en la selección de características.

Como vimos, se realizaron dos conjuntos de experimentos para comparar los resultados de un AG aplicado a los datos originales frente a su aplicación a datos aumentados:

1. Datos originales: Se trabajó directamente con las muestras disponibles en el conjunto Leukemia, siguiendo la partición original en un conjunto de entrenamiento y un conjunto de prueba.
2. Datos aumentados: Se realizaron experimentos con 100, 200 y 1000 muestras sintéticas adicionales (a las 38 originales) mediante un AV entrenado específicamente para este conjunto de datos.

La primera serie de experimentos con datos originales y datos aumentados tuvo por objetivo contar con una primera aproximación al problema y establecer una línea base a partir de la cual iterar con ajustes y mejoras en la configuración del AG y el AV. Luego, se realizaron experimentos adicionales para explorar diferentes configuraciones del AG y el AV, con el objetivo de identificar las condiciones óptimas para la selección de características en este conjunto de datos. Se investigó el impacto de variar la probabilidad de mutación, la probabilidad de cruce, la proporción de genes activos en el cromosoma en la inicialización y el número de muestras generadas por el AV. Particularmente se exploró el impacto de la reducción de la proporción de genes activos en el cromosoma en la eficiencia de la selección de características, pasando de p = 0.1 (aproximadamente 712 características), 0.01 (71), y 0.005 (35).

En el primer conjunto de experimentos, se estableció una configuración base utilizando una probabilidad de mutación de 0.01, una probabilidad de cruce del 0.75, y una proporción de genes activos en el cromosoma del 10 % del total de características, limitado a un máximo de 20 generaciones. Esta configuración fue seleccionada para equilibrar la exploración y explotación del espacio de búsqueda, asegurando que el AG pudiera explorar adecuadamente las posibles combinaciones de características sin prologar la búsqueda en exceso. Para los experimentos con aumentación de datos, se generaron 100 muestras sintéticas adicionales mediante un AV.

En estas primeras pruebas los resultados mostraron que tanto la exactitud como el número de genes seleccionados fueron similares entre los grupos con y sin aumentación de datos. Sin embargo, se destacó una menor dispersión en la variaza de la exactitud en los resultados del grupo con aumentación, lo que sugirió que la generación de datos sintéticos contribuyó a una mayor estabilidad del modelo.

En un segundo grupo de experimentos exploratorios, se investigaron variaciones en la configuración de la proporción de genes activos en el cromosoma y el tamaño del conjunto de datos, para examinar en profundidad los efectos de la aumentación. Específicamente, se realizaron pruebas con conjuntos de datos aumentados que incluían 200 y 1000 muestras sintéticas adicionales, y se redujo agresivamente la proporción de genes activos en el cromosoma, en algunos casos hasta un 0.5 % de las características totales. Estas configuraciones más extremas fueron seleccionadas para evaluar la robustez del AG frente a diferentes tamaños del espacio de búsqueda, especialmente en escenarios donde se esperaba que la reducción dimensional comprometiera la capacidad del AG para encontrar buenas soluciones.

Resultados

En los experimentos realizados sobre el conjunto de datos Leukemia, los resultados mostraron leves diferencias en favor de la aumentación en lo que respecta a exactitud.

En la Figura 4.7 cada punto representa la exactitud promedio de la población al finalizar el proceso evolutivo en un experimento (1 punto = 1 experimento). Aunque la diferencia no es significativa, sí se observa una mayor estabilidad (menor varianza) en los resultados del grupo con datos aumentados.

Cabe aclarar que dada las distintas configuraciones de cromosoma empleadas en los experimentos, el gráfico referido a genes aparece marcadamente dividido en dos grupos: experimentos donde los genes de los individuos tenían $< 100$ características activas (abajo), y experimentos con ~700 características activas (arriba). Pese a constituir experimentos claramente diferentes, incluimos ambos en la misma representación pues no observamos diferencias relevantes en la cantidad de genes activos en los grupos con aumentación frente a los originales.

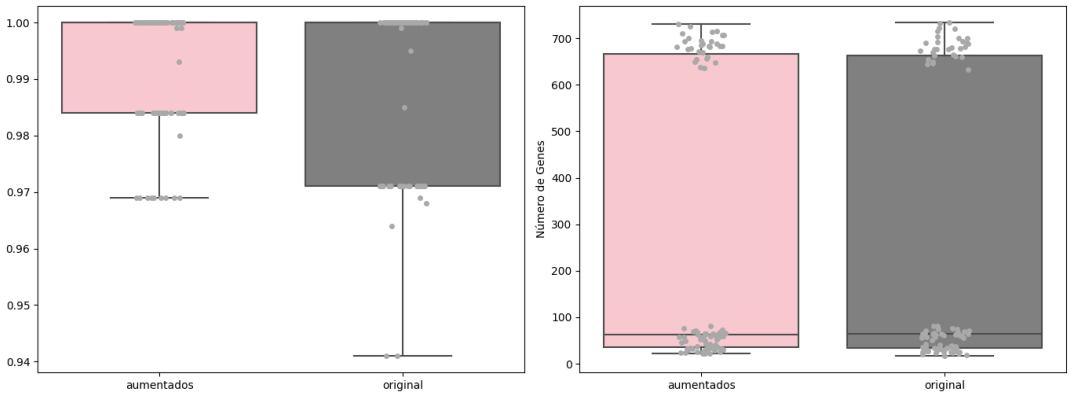


Figura 4.7 Exactitud (izquierda) y número de características seleccionadas (derecha) en Leukemia. Datos aumentados (rosado) y datos originales (gris). Los resultados se evaluaron en datos de testeo sin aumentación.

Entendemos posible que, en conjuntos de datos donde los modelos alcanzan una exactitud alta (como sucede en este caso con Leukemia, pero también sucede con Gisette, que veremos en breve), la aumentación no produce diferencias sustanciales en la performance del AG.

Un punto a resaltar sobre este dataset es la alta correlación entre las características, lo que se refleja en la cantidad significativa de características seleccionadas al menos una vez durante todas las pruebas realizadas. Tanto en los experimentos con datos aumentados como en los originales, el AG seleccionó al menos una vez un número muy similar de características (6779 con datos aumentados y 6772 con datos originales), cubriendo así una fracción alta del espacio de búsqueda. Este comportamiento sugiere que la mayoría de las características están fuertemente vinculadas, lo que permite que el AG identifique soluciones efectivas utilizando diferentes subconjuntos de características. Asimismo, se observó que un pequeño porcentaje de características (aproximadamente el 10 %) es suficiente para resolver el problema de clasificación, independientemente de cuáles sean esas características, debido a la redundancia en la información aportada por las correlaciones. El análisis cuantitativo de las correlaciones muestra que, aunque solo el 0.32 % de los pares de características posibles tienen una correlación absoluta mayor a 0.7, estos representan un número considerable (80742) de correlaciones significativas. Es decir: aunque el número de características altamente correlacionadas es una fracción pequeña del total, su impacto en la selección de características es notable, dado que el AG tiende a seleccionar conjuntos de características que son efectivamente intercambiables debido a su alta redundancia.

## 4.10. Gisette

Como se mencionó en el Capítulo 2, este conjunto de datos fue seleccionado para evaluar cómo la aumentación de datos afecta la selección de características en un contexto donde el espacio de características es grande, pero la relación señal-ruido es moderada.

La metodología seguida en los experimentos con Gisette fue similar a la utilizada en Leukemia. La configuración del AG estuvo establecida en: probabilidad de mutación de 0.0002, probabilidad de cruce de 0.75, proporción de genes activos en el cromosoma de 0.1 y un número máximo de generaciones de 30. Se prestó especial atención a la reducción del espacio de búsqueda mediante una proporción de genes activos en el cromosoma equivalente a 500 características. Se investigó también el impacto de generar un número mucho mayor de muestras sintéticas (6000).

Resultados

Los resultados en Gisette fueron similares a los observados en Leukemia. La exactitud media en el conjunto de datos de testeo fue ligeramente superior en los experimentos con datos aumentados (0.960) en comparación con los datos originales (0.959), pero sin diferencia significativa.

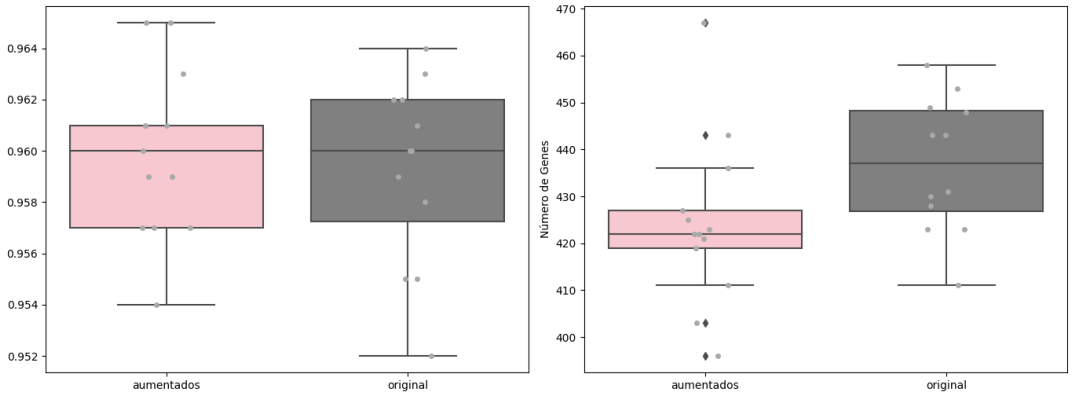


Figura 4.8. Exactitud (izquierda) y número de características seleccionadas (derecha) en Gisette. Datos aumentados (rosado) y datos originales (gris).Los resultados se evaluaron en datos de testeo sin aumentación.

Al igual que en el caso anterior este hallazgo parece sugerir que la aumentación de datos no produce diferencias significativas en la performance del AG en conjuntos de datos donde los modelos ya alcanzan una alta exactitud, y donde la cantidad de observaciones originales es suficiente para la aplicación de estrategias de selección. Sin embargo, la estabilidad de la selección de características sí parece sugerir una mejora con la aumentación, como se refleja en la menor desviación estándar y el rango intercuartílico en la Tabla 4.2.

| | Datos aumentados | Datos originales |
|---|---|---|
| Características seleccionadas (media) | 424 | 436 |
| Desviación Estándar | 17 | 14 |
| Rango Intercuartil | 8 | 21 |

Tabla 4.2. Comparación de experimentos en Gisette

En efecto, en Gisette se observa una diferencia más importante en la estabilidad de la selección de características. En los experimentos con datos aumentados, a lo largo de las diferentes corridas del algoritmo propuesto, el número promedio de características seleccionadas fue menor (424) en comparación con los datos originales (436), señalando que el AG fue más efectivo en el reconocimiento de un subconjunto relevantes.

## 4.11. Madelon

El conjunto de datos Madelon es un caso especial donde solo cinco características son relevantes, y las otras quince son combinaciones lineales de estas, y el resto son datos aleatorios. Así, este conjunto representa un desafío interesante para evaluar la capacidad de los AG para identificar características útiles en un entorno donde la señal está oculta entre una gran cantidad de ruido.

La metodología seguida en los experimentos con Madelon fue similar a la utilizada en Leukemia y Gisette: en todos los casos supuso una serie de experimentos exploratorios para determinar las mejores configuraciones de parámetros. En el caso que nos ocupa, dicha configuración del AG fue la siguiente: la probabilidad de mutación se estableció en 0.002, la probabilidad de cruce en 0.75 y la proporción de genes activos en el cromosoma en 0.1. El número máximo de generaciones se estableció en 30.

Se generaron 2000 muestras sintéticas para incrementar el número de observaciones disponibles y evaluar si esto mejoraba la capacidad del AG para encontrar las características relevantes. Como en los experimentos precedentes, se exploraron diferentes configuraciones de la proporción de genes activos en el cromosoma al momento de la inicialización, y se investigó cómo la aumentación de datos, en el contexto de un espacio de búsqueda complejo y ruidoso, afectaba la eficiencia de la selección de características.

Resultados

Los experimentos en Madelon mostraron resultados significativamente diferentes a Leukemia y Gisette como puede apreciarse en la Figura 4.9. La exactitud media en los experimentos con datos aumentados fue de 0.83, lo que representa un aumento del 10.4% en comparación con la exactitud de los datos originales de 0.75. Esta diferencia es significativa, lo que indica que la aumentación de datos tuvo un impacto positivo en el desempeño del AG.

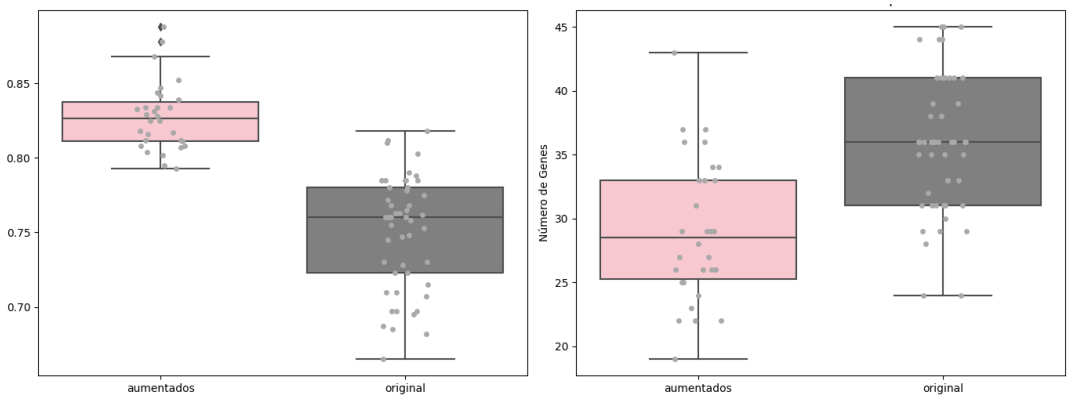


Figura 4.9. Exactitud (izquierda) y número de características seleccionadas (derecha) en Madelon. Datos aumentados (rosado) y datos originales (gris).Los resultados se evaluaron en datos de testeo sin aumentación.

Estos resultados sugieren que la aumentación de datos puede ser especialmente efectiva en conjuntos de datos con características complejas y un alto nivel de ruido, como es el caso de Madelon. La generación de muestras sintéticas permitió al AG identificar mejor las características relevantes, lo que se tradujo en una mejora significativa en la exactitud del modelo.

Por otro lado, el análisis de selección de características reveló que, en promedio, el número de características seleccionadas fue menor en los experimentos con datos aumentados (29) en comparación con los datos originales (35). Este hallazgo, también resalta el efecto positivo que produce la aumentación de datos en la selección de características.

Finalmente, podemos destacar también que de los clasificadores clásicos evaluados en el Capítulo 2, que no tuvieron el tratamiento que aquí estamos analizando (aumentación + selección de características), solo 2 de los 18 logran valores de exactitud superiores a los que aquí estamos presentando. Ambos modelos, AdaBoost y Bagging, lograron esos resultados luego de un proceso de optimización de hiperparámetros. El resto de los modelos clásicos oscila entre 0.5 y 0.7 de exactitud, lo que refuerza la idea de que la aumentación de datos y la selección de características puede ser una estrategia efectiva para mejorar los resultados de un clasificador en conjuntos de datos complejos.

## 4.12. GCM

El conjunto de datos GCM consiste en 16063 atributos (biomarcadores), sobre 190 muestras de tumores que refieren a 14 clases de cáncer humano. Este problema representa un desafío aún mayor que los anteriores no solo debido a la alta dimensionalidad y bajo número de muestras, sino también por las múltiples clases y el desbalance entre ellas.

El proceso de experimentación en GCM estuvo dividido en dos partes. En la primera parte, se realizaron experimentos exploratorios con diferentes configuraciones del AVC y el AG. El objetivo de esta etapa fue establecer una línea base para evaluar la efectividad de la aumentación de datos. En la segunda parte, se realizaron ajustes en la metodología, el modelo de AVC y el diseño de la integración AVC-AG. Esta vez, el objetivo era mejorar la calidad de los datos sintéticos y la eficiencia de la selección de características.

Aunque se realizaron una gran cantidad de experimentos (algunos exploratorios y otros más específicos), nos centraremos en los resultados de la primera y segunda parte que se describen a continuación, que nos permitieron identificar los desafíos y oportunidades en la utilización de AV en conjuntos de datos complejos como GCM.

| n_muestras | cantidad_experimentos |
|---|---|
| 133 | 20 |
| 254 | 2 |
| ~1300 | 10 |
| 3129 | 15 |
| 6106 | 1 |

Tabla 4.3. Cantidad de experimentos por cantidad de muestras sintéticas

La cantidad de muestras sintéticas generadas en los experimentos fue ajustada a lo largo de las pruebas, conforme se exploraban opciones de mejora. Ello dio lugar a los valores de la Tabla 4.3, donde detallamos cantidad de experimentos y la cantidad de muestras sintetizadas que se emplearon en cada uno. La primera fila representa la cantidad experimentos con las muestras originales (sin aumentación).

### 4.12.1. Primera etapa de experimentación

Al enfrentarnos a GCM asumíamos que la generación sintética de muestras y ulterior proceso de selección de características plantearía un desafío significativo al AG debido a la complejidad del dataset. La calidad de reconstrucción lograda en los datos sintéticos generados por el AVC presentado en el Capítulo 3, aunque capaz de preservar la estructura subyacente del conjuntos de datos de GCM, no permitía suponer una mejora significativa en la exactitud de la clasificación con un AG. Por otro lado, el desbalance que tienen las clases en GCM, determinaba que la calidad de reconstrucción de las clases minoritarias fuera menor, lo que podría afectar la capacidad del AG para identificar un subconjunto relevante de características. Por todo ello, esperábamos que los experimentos con datos aumentados presentaran resultados mixtos, con posibles mejoras en la estabilidad de la selección de características, pero sin mejoras significativas en la exactitud.

Respecto de los recursos de hardware utilizados, dado que la aumentación de datos requeriría un poder de cómputo proporcional al tamaño del dataset por la cantidad de muestras sintéticas generadas, se optó por utilizar un entorno en la nube para ejecutar los experimentos de manera eficiente. Para este propósito, se trabajó con una máquina virtual con 8 vCPUs y 30 GB de RAM.

La configuración inicial del AG fue la siguiente: se establecieron probabilidades de mutación en 0.0006, 0.01 y 0.1, probabilidad de cruce en 0.75, proporción de genes activos en el cromosoma en 0.1 y un número máximo de generaciones de 20.

En la primera etapa se experimentó con una proporción de genes activos en el cromosoma que representaba el 10 % de las características totales, alrededor de 1600 genes por cromosoma, tanto para el grupo de experimentos con datos originales como al grupo de experimentos con datos aumentados. Esta medida se adoptó luego de realizados dos experimentos exploratorios con cromosomas activos en valores del 20 % y 30 % de las características totales, donde se observó una acentuada degradación en la exactitud (en el rango de 0.2 y 0.35).

Asimismo se varió la probabilidad de mutación en tres valores: 0.0006, 0.01 y 0.1, para instar una exploración más amplia del espacio de búsqueda. La probabilidad de cruce se mantuvo constante en 0.75, y el número máximo de generaciones fue de 20.

Respecto del grupo de experimentos con datos aumentados, el dataset mixto de entrenamiento incluía 1400 muestras sintéticas (100 instancias por clase) y la partición original de entrenamiento (con 133 observaciones). La evaluación se realizó sobre los datos originales de testeo (57 observaciones).

Los resultados, como anticipabamos, no fueron favorables. Como puede verse en la Figura 4.10 la exactitud media fue ligeramente superior en los experimentos con datos originales (0.538) en comparación con los datos aumentados (0.512). La estabilidad de la selección de características fue similar en ambos grupos, con una dispersión similar en la cantidad de características seleccionadas, como se puede observar en el siguiente gráfico.

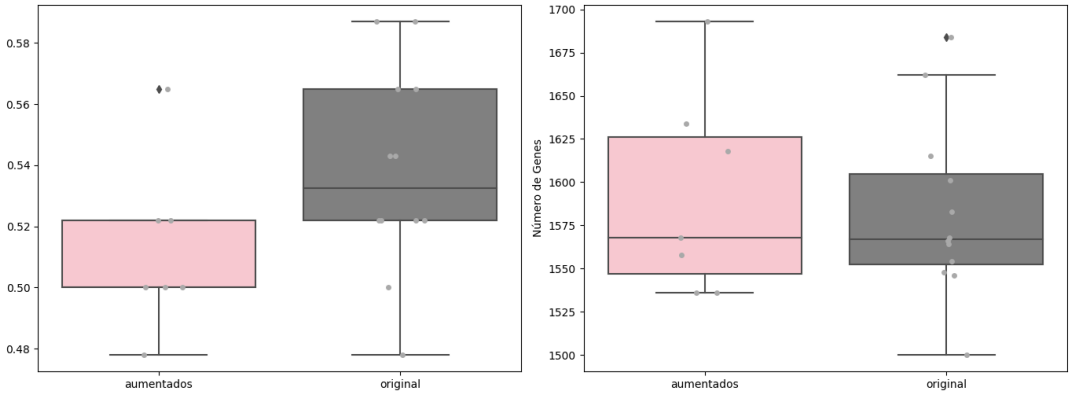


Figura 4.10 Exactitud (izquierda) y número de características seleccionadas (derecha) en GCM, primera etapa. Datos aumentados (rosado) y datos originales (gris). Los resultados se evaluaron en datos de testeo sin aumentación.

Como resultado de esta primera etapa de experimentos se identificó una dificultad importante en la metodología utilizada. La calidad de los datos sintéticos generados por el AVC no contribuyó a mejorar la exactitud de la clasificación, consecuentemente el AG con datos aumentados no tiene mejores resultados que el AG con datos originales. Ante esta circunstancia se procedió a realizar un test de diagnóstico para evaluar la calidad de los datos sintéticos. Se generaron 3000 muestras, se entrenó y evaluó un MLP con estos datos. Los resultados en la partición sintética de testeo fueron los siguientes:

| Clase | Precisión | Recall | F1-Score | Soporte |
|---|---|---|---|---|
| 0 | 1.00 | 1.00 | 1.00 | 23 |
| 1 | 1.00 | 1.00 | 1.00 | 28 |
| 2 | 1.00 | 1.00 | 1.00 | 31 |
| 3 | 1.00 | 1.00 | 1.00 | 31 |
| 4 | 0.97 | 1.00 | 0.98 | 32 |
| 5 | 1.00 | 0.97 | 0.99 | 34 |
| 6 | 0.96 | 1.00 | 0.98 | 26 |
| 7 | 1.00 | 0.97 | 0.98 | 31 |
| 8 | 1.00 | 1.00 | 1.00 | 30 |
| 9 | 1.00 | 1.00 | 1.00 | 38 |
| 10 | 1.00 | 1.00 | 1.00 | 30 |
| 11 | 1.00 | 1.00 | 1.00 | 31 |
| 12 | 1.00 | 1.00 | 1.00 | 26 |
| 13 | 1.00 | 1.00 | 1.00 | 29 |
| Exactitud | | | 1.00 | 420 |
| Macro avg | 1.00 | 1.00 | 1.00 | 420 |
| Weighted avg | 1.00 | 1.00 | 1.00 | 420 |

Tabla 4.4. Resultados del MLP en los datos sintéticos de GCM

Estos resultados sugieren que el MLP fue capaz de aprender correctamente la estructura de los datos sintéticos, lo que se tradujo en una exactitud del 100 % en la clasificación, sin embargo, el mismo modelo evaluado en los datos de test originales arrojó una exactitud de 0.5. Esta discrepancia sugiere que la distribución de probabilidad de los datos sintéticos no es representativa de la que caracteriza a los datos

reales, circunstancia que estaría limitando la capacidad del AG para identificar un subconjunto relevante de características. Esto explicaría los motivos por los cuales los experimentos con datos aumentados no lograron mejorar la exactitud de la clasificación.

En este punto, asumimos que un problema importante que tenía nuestro modelo de AVC se explicaba por efecto de la función de pérdida sobre la reconstrucción, en lo que se conoce como colapso de la distribución posterior [2]. En efecto, como vimos en el Capítulo 3, dicha función resulta de la combinación de dos términos: la pérdida de reconstrucción y la divergencia KL. La divergencia KL es una medida de la diferencia entre dos distribuciones de probabilidad, y se utiliza para regularizar el espacio latente, mantenerlo distribuido y compacto; mientras que la pérdida de reconstrucción mide la diferencia entre los datos originales y los datos reconstruidos. En el caso de GCM, la divergencia KL podría estar dominando la función de pérdida, lo que resultaría en una distribución latente regularizada, pero no necesariamente representativa de los datos reales. Esto significa que el modelo está produciendo una distribución convenientemente discriminada, a expensas de la exactitud en la reconstrucción de los datos originales. Situación particularmente problemática cuando los datos presentan relaciones complejas como en GCM. Ante esta problemática, entendimos que se abrían distintos caminos: podíamos ajustar los pesos de la divergencia KL y la reconstrucción para que el AVC sea capaz de generar datos que sean representativos de los datos reales, o bien, explorar la aplicación del AVC a un subespacio de características seleccionadas por un AG, lo que podría reducir la complejidad del problema y mejorar la calidad de los datos sintéticos. En la próxima parte de los experimentos, exploramos esta segunda opción.

### 4.12.2. Segunda etapa de experimentación

Para abordar las limitaciones observadas, se realizaron ajustes en distintas configuraciones.

En primer lugar, se adaptó la función de pérdida para incluir un factor de peso por clase que permitiera rebalancear el desequilibrio entre las clases. Ello con el fin de penalizar con mayor severidad los errores en las clases minoritarias, y así procurar una representación más fiel de la distribución real de los datos.

En segundo lugar, se integró un muestreador aleatorio ponderado en la etapa de entrenamiento del AVC, que permitiría ajustar los lotes de entrenamiento con igual objetivo de mejorar la representación de las clases.

Finalmente, se ajustó la etapa de inicio del proceso generativo, aplicando el AVC ya no al conjunto de datos completo, sino a un subconjunto de características seleccionadas por un AG. La idea era reducir la complejidad del problema y mejorar la calidad de los datos sintéticos, permitiendo al AVC enfocarse en un espacio de características reducido y relevante. Veamos esto último en detalle.

[2] El fenómeno de colapso de la distribución posterior se manifiesta cuando la distribución posterior aproximada, derivada de la red codificadora a partir de los datos de entrada, converge a la distribución previa (en nuestro caso una gaussiana estándar) durante el entrenamiento de un AV. Esto ocurre debido al término de divergencia de Kullback–Leibler incluido en el ELBO, el cual impone una regularización que penaliza la desviación de la distribución posterior respecto a la distribución previa (Dang et al. 2024). El resultado es que el modelo se adapta a la distribución previa, produciendo una distribución posterior que no es representativa los datos reales.

Como vimos, una de las limitaciones principales identificadas en los experimentos iniciales fue la dificultad del AVC para generar datos sintéticos que representaran fielmente la distribución de los datos reales. Para abordar este problema, planteamos una hipótesis: si aplicábamos el AVC a un subconjunto seleccionado de características relevantes (en lugar de todas las características), podríamos simplificar la distribución de probabilidad que el modelo necesitaba aprender, y de esa manera podría mejorar la calidad de los datos sintéticos generados.

Para verificar esta hipótesis, diseñamos un flujo de trabajo en tres etapas:

1. Selección inicial: Primero identificamos un subespacio de características relevantes utilizando los resultados de experimentos previos con AG. Este era un paso crítico porque debía reducir el espacio del problema pero al mismo tiempo evitar pérdida de información relevante. Entonces, para lograr esos objectivos, seleccionamos las 1600 características que aparecían con mayor frecuencia en los 590 cromosomas obtenidos de experimentos previos, donde cada cromosoma representaba aproximadamente el 10 % de las características totales (16063).

2. Generación: Este subconjunto preseleccionado de características se utilizó como entrada para el AVC, que ahora podía enfocarse en aprender y modelar un espacio de menor dimensionalidad. El modelo generativo entrenado con este subespacio produjo datos sintéticos potencialmente de mayor calidad.

3. Selección final: Finalmente, aplicamos nuevamente un AG para realizar la selección de características, pero esta vez utilizando los datos aumentados (originales + sintéticos) generados en la etapa anterior.

Este enfoque secuencial de "selección-generación-selección" permitió concentrar el aprendizaje de los modelos en un subespacio de características más relevante, facilitando que el AVC capturara mejor las relaciones importantes entre variables y generara datos sintéticos más representativos de la distribución original.

Respecto de la configuración de los experimentos se implementaron las siguientes modificaciones: se ajustó el total de muestras sintéticas generadas por el AVC a 3000. Se redujo la proporción de genes activos en el cromosoma a 0.05 y 0.025 respecto del espacio original de características (~750 y ~450 atributos respectivamente). Se mantuvo constante la probabilidad de cruce en 0.75, como también la probabilidad de mutación en 0.1 (~0.0006), en ambos casos siguiendo la configuración con mejores resultados en nuestros experimentos exploratorios. El número máximo de generaciones se redujo a 15, considerando la reducción en la complejidad del problema y luego de constatar que el algoritmo lograba convergencia dentro de esa cantidad de generaciones.

Como resultado de todas estas modificaciones, se esperaba una mayor calidad de los datos sintéticos, lo que se traduciría en una mejora en la exactitud de la clasificación, y en la eficiencia de la selección de características.

Resultados Parte 2

Efectivamente, los experimentos realizados en la segunda parte de la investigación mostraron una mejora significativa en la exactitud de la clasificación.

A pesar del desafío que supone la clasificación de las 14 clases desbalancedadas que posee el dataset de testeo de GCM, la exactitud media en los experimentos con datos aumentados fue de 0.541, lo que representa un aumento de 8.85 % en comparación con la exactitud de los datos originales de 0.497. Esta diferencia es significativa, lo

que indica que la aumentación de datos tuvo un impacto positivo en el desempeño del AG.

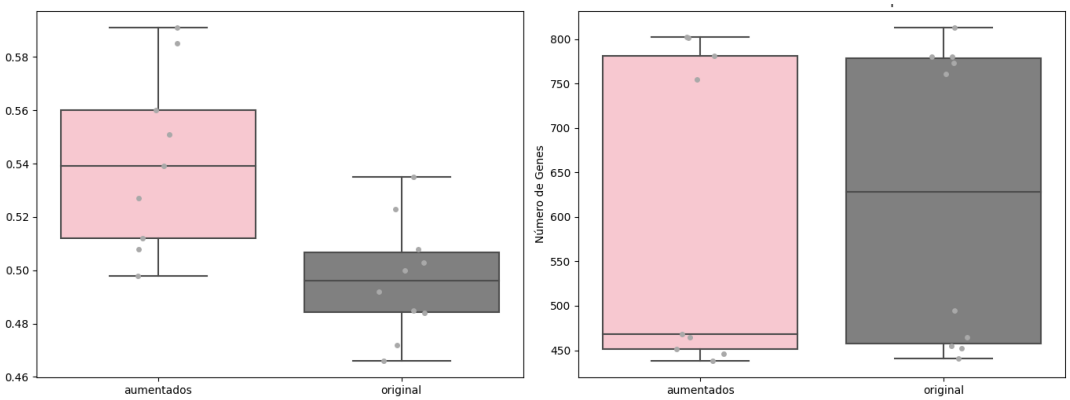


Figura 4.11. Exactitud (izquierda) y número de características seleccionadas (derecha) en GCM, segunda etapa. Datos aumentados (rosado) y datos originales (gris).Los resultados se evaluaron en datos de testeo sin aumentación.

Respecto a la cantidad de características seleccionadas, no se observa gran diferencia entre los experimentos con datos originales y los datos aumentados. En efecto, el número de características promedio seleccionadas en los experimentos con aumentación y cromosomas de tamaño 0.05 y 0.025 fue de 453 y 784 respectivamente, en comparación con 461 y 781 en los experimentos con datos originales.

Para validar los resultados obtenidos, se realizó un grupo de experimentos adicionales donde se disminuyó la proporción de genes activos en el cromosoma a 0.015 y 0.003, lo que representó una selección de atributos en torno a ~200 y ~45. Los resultados de la Figura 4.12 mostraron que, pese a la importante reducción en el espacio de búsqueda, la exactitud se mantuvo en un nivel aceptable en ambos grupos. Los mejores resultados se obtuvieron con datos aumentados, a excepción del grupo de experimentos con un tamaño del cromosoma activo de ~200 características.

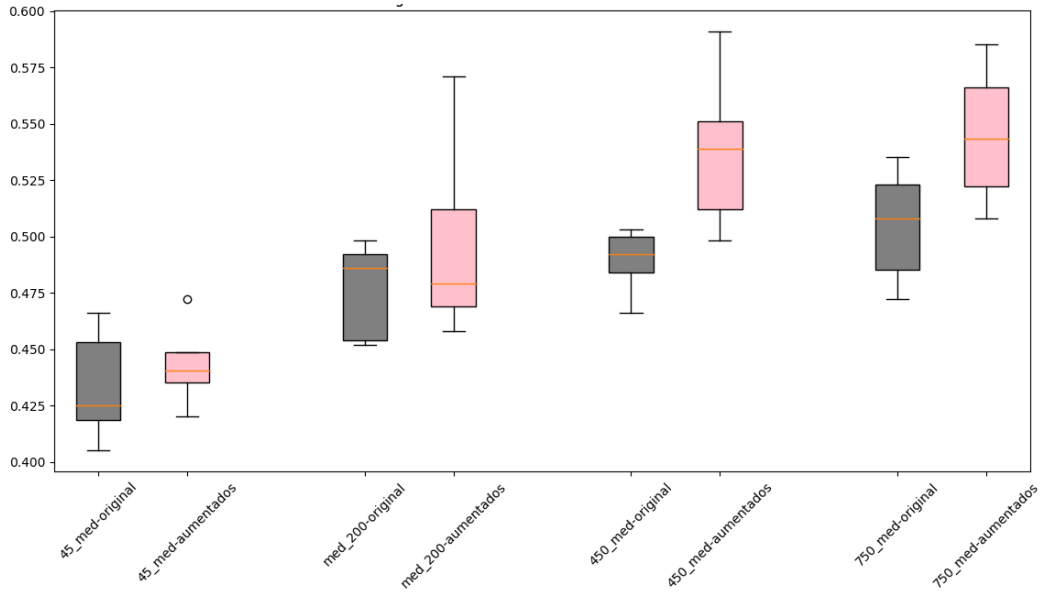


Figura 4.12 Experimentos con GCM variando el número de características seleccionadas. Datos aumentados (rosado) y datos originales (gris).

A continuación presentamos la serie completa de experimentos realizados en GCM, donde se puede observar la evolución de la exactitud en función de la proporción de genes activos en el cromosoma. La Figura 4.12 muestra los cuatro grupos de experimentos con sus correspondientes resultados en términos de exactitud, tanto para datos aumentados (rosado) como datos sin aumentación (gris). Cada grupo de experimentos posee una distinta configuración de genes activos (promedio), que se inicia con ~45 en el primer grupo, hasta llegar a los ~750 en el último grupo.

Finalmente, se realizó un experimento con 6000 muestras sintéticas, donde se observó una degradación de los resultados. Esto sugiere que la generación de un número excesivo de muestras sintéticas puede comprometer la calidad de los datos y afectar negativamente la exactitud de la clasificación.

Todo lo anterior sugiere que, si bien la aumentación de datos mediante AVC puede enfrentar desafíos significativos en conjuntos de datos complejos como GCM, es posible mejorar la calidad de los datos sintéticos mediante ajustes en la arquitectura del AVC y en la metodología de selección de características. La combinación de AVC y AG en un flujo de trabajo encadenado demostró ser una estrategia prometedora para mejorar la exactitud y la eficiencia en la selección de características.

## 4.13. ALL Leukemia

Con el fin de validar el tratamiento dado a GCM, particularmente la estrategia de encadenamiento de procesos de selección-generación-selección, se decidió aplicar la misma metodología a un nuevo dataset All-Leukemia. Este dataset, que contiene 12600 variables (i.e. genes) y 327 muestras, es un caso de estudio clásico en la literatura de clasificación de tumores.

Siguiendo la metodología implementada en los 4 conjuntos de datos precedentes, se repitió el procedimiento de búsqueda de la mejor combinación de hiperparámetros para el AVC, con el fin de lograr la mejor reconstrucción de los datos originales. Por su parte, se mantuvo la configuración del modelo AVC utilizado en GCM, descripto en el Capítulo 3, consistentes en 3 capas en las redes de codificación y decodificación.

Al igual que en GCM, se realizaron experimentos con datos originales y aumentados. Los experimentos con aumentación consistieron en la generación de 1000 muestras sintéticas creadas por un AVC entrenado en un subespacio de características relevantes, seleccionadas por un AG. Esta selección del subespacio se llevó a cabo extrayendo las características más frecuentes a lo largo de 3 experimentos de selección. Tanto para la generación como para la selección siempre se trabajó con las particiones de datos originales de entrenamiento y testeo.

Se probaron 3 escenarios de reducción de características, con un ratio de genes activos de 0.003, 0.015 y 0.025, dando como resultado un tamaño de cromosoma en torno a las 35, 180 y 300 variables, respectivamente. El algoritmo genético utilizado en esta oportunidad realizaba 15 generaciones.

Los resultados generales se pueden observar en la Figura 4.13.

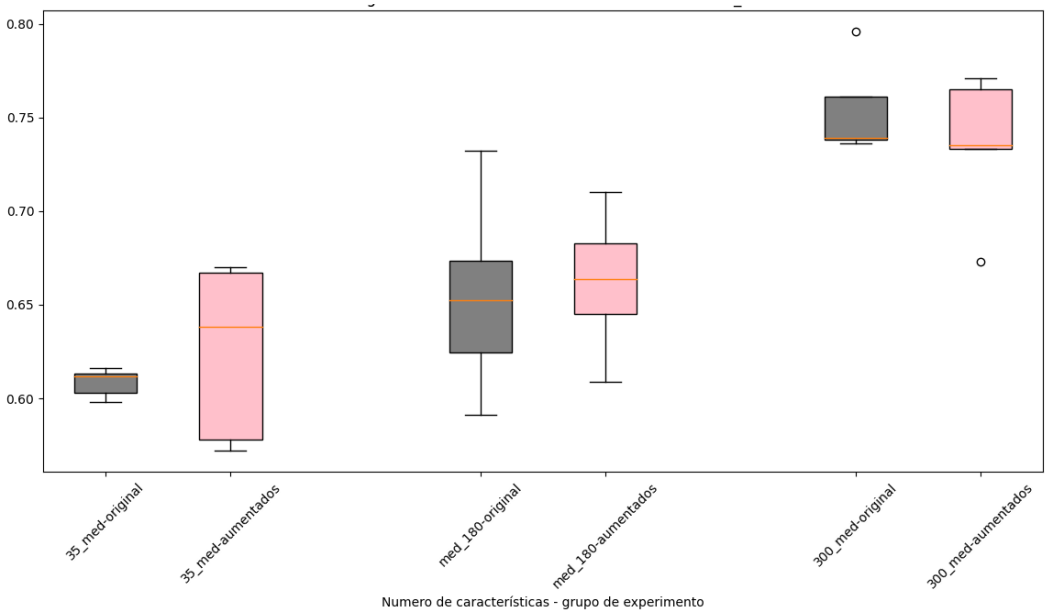


Figura 4.13 Experimentos con ALL-Leukemia variando el número de características seleccionadas. Datos aumentados (rosado) y datos originales (gris). Los resultados se evaluaron en datos de testeo sin aumentación.

Como puede advertirse en la Figura 4.13, los experimentos muestran que la estrategia de encadenamiento de procesos de selección-generación-selección presenta una ligera mejora en la exactitud de clasificación en los casos de drástica reducción de características en torno a los 35 y 200 genes activos. No se evidencia diferencia significativa entre los experimentos con 300 genes activos.

Respecto de la eficacia en la selección de características, no se observa una diferencia significativa entre los experimentos con datos originales y los datos aumentados como puede apreciarse en los siguientes gráficos.

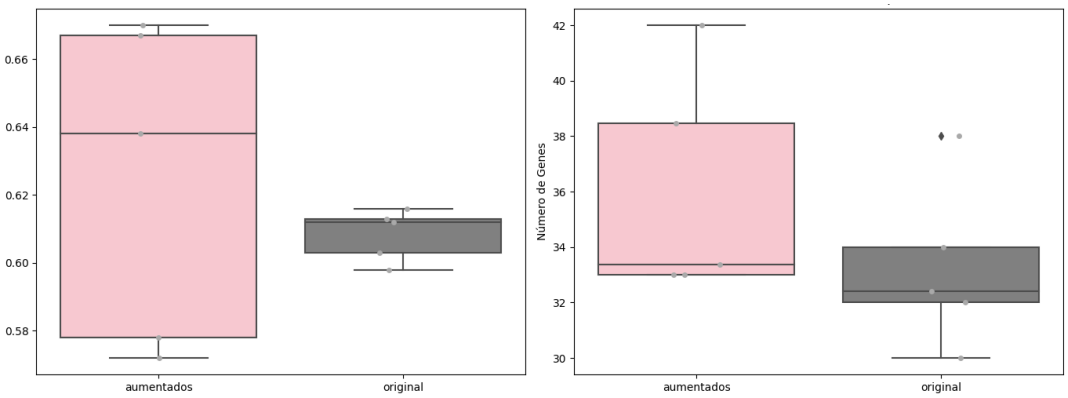


Figura 4.14. Resultados de All-Leukemia con 35 genes activos. Datos aumentados (rosado) y datos originales (gris).

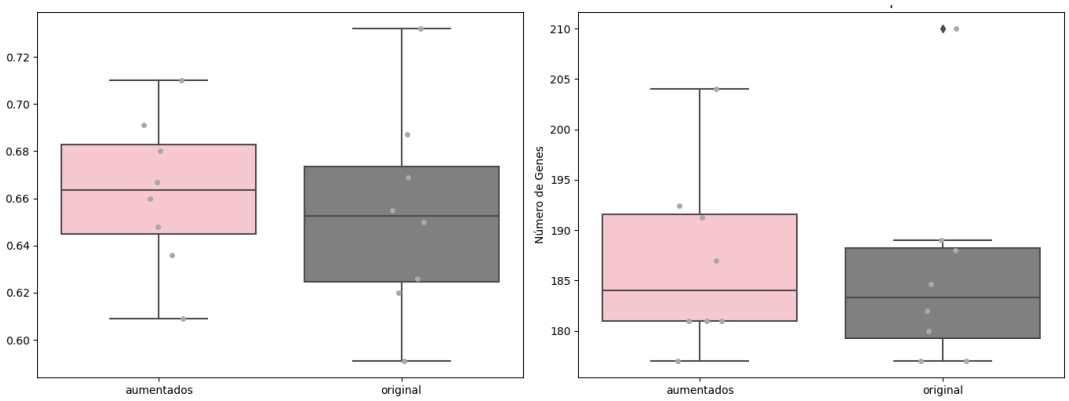


Figura 4.15. Resultados de All-Leukemia con 180 genes activos. Datos aumentados (rosado) y datos originales (gris).

En el caso del experimento con 35 genes activos (promedio) se advierte, pese a la mejora en la exactitud, una leve degradación en la estabilidad de los resultados. Entendemos que esto se explica por la reducción en la cantidad de genes activos y la sensibilidad del AG a parámetros como la probabilidad de mutación. En este caso, el experimento tuvo una probabilidad de mutación de 0.028, operando una importante variabilidad en la estructura de los cromosomas y afectando la selección.

## 4.14. Resumen de los resultados

Los experimentos realizados en los primeros cuatro conjuntos de datos (Leukemia, Gisette, Madelon y GCM) evidencian resultados positivos sobre el impacto de la aumentación de datos mediante Autocodificadores Variacionales en la selección de características utilizando Algoritmos Genéticos. Los resultados generales puede apreciarse en las Figuras 4.16 y 4.17, donde se presentan los valores obtenidos para exactitud y cantidad de genes activos luego del proceso evolutivo del AG. Esta tendencia positiva tiene confirmación en los resultados obtenidos en el quinto dataset All-Leukemia.

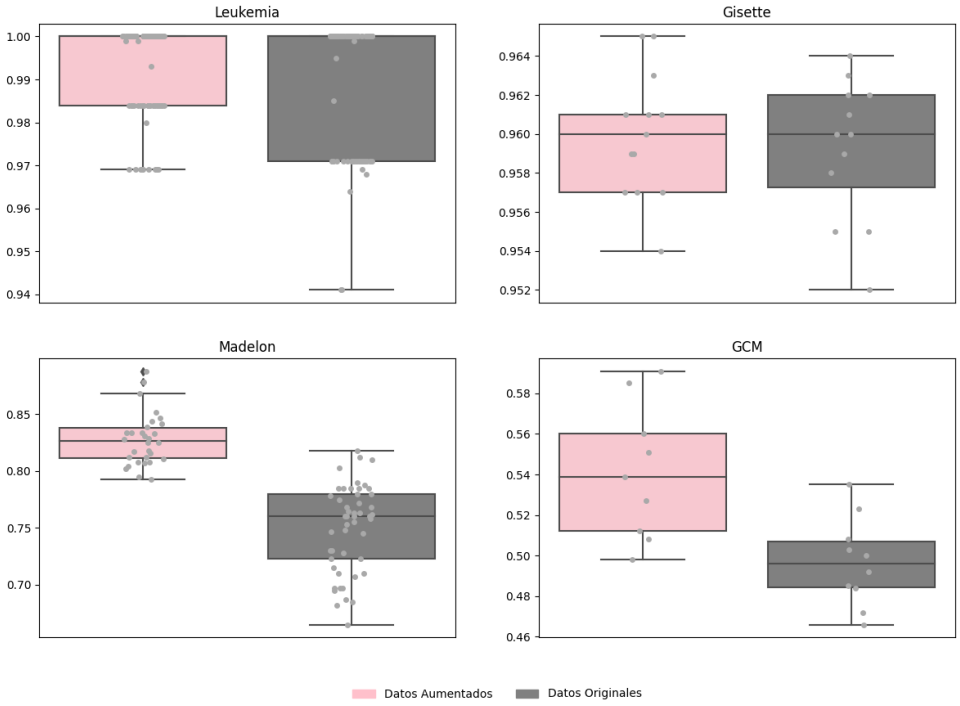


Figura 4.16 Exactitud en los 4 datasets.

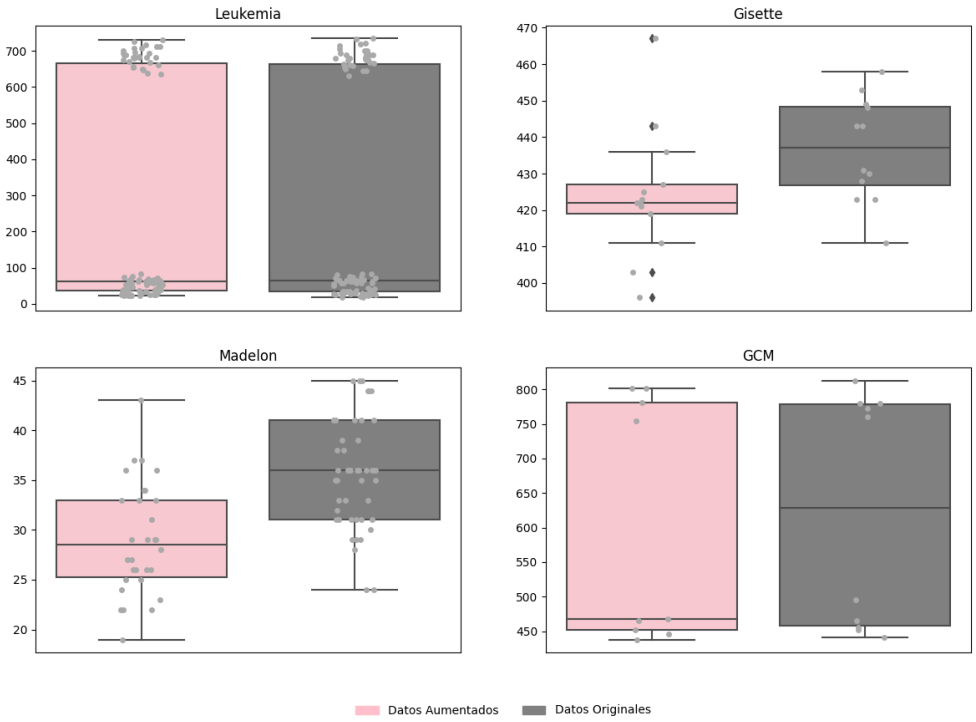


Figura 4.17 Número de características seleccionadas en los 4 datasets.

El uso de datos aumentados no siempre produjo mejoras en la exactitud de clasificación, pero sí marcó una diferencia relevante ante los casos más difíciles. En efecto, en problemas con métricas saturadas su aporte agrega un valor marginal, asociado a la estabilidad de los resultados, como se vió en Leukemia y Gisette. Ambos datasets

representan desafíos donde los modelos ya alcanzan una alta performance procesando los datos en su estado original. Sin embargo, en dataset más complejos, donde el margen de mejora en las predicciones es mayor, como los casos de Madelon, GCM y All-Leukemia, la técnica de aumentación generó resultados positivos, particularmente en los dos primeros donde se observó un salto del 10 % y 9 % respectivamente en el resultado final. Este hallazgo sugiere que la aumentación de datos puede contribuir a generar modelos más consistentes a lo largo de múltiples generaciones, lo cual es particularmente relevante en escenarios de alta dimensionalidad, escasez muestral y ruido.

En el caso del conjunto de datos Madelon, donde el problema de selección de características presenta una clara distinción entre datos relevantes y ruido, la aumentación de datos mostró una mejora significativa en la exactitud, incrementando la capacidad del AG para identificar las características correctas en un espacio de búsqueda extremadamente ruidoso. Este resultado resalta la utilidad de la aumentación en problemas donde la presencia de señales distractoras dificulta la tarea de selección.

Por otro lado, los experimentos con GCM ilustran las limitaciones de la aumentación en conjuntos de datos con una distribución de clases desbalanceada y de alta complejidad. Sin embargo los ajustes en la arquitectura del AVC y en la configuración experimental permitieron mejoras importantes, alcanzando una generación de datos sintéticos de mayor calidad y una selección de características más eficiente. Resultados que se validaron mediante los experimentos en All-Leukemia. En este sentido, la integración de estrategias más complejas, como la combinación de selección de características previa a la aumentación, demostró ser una vía prometedora para mejorar los resultados.

En conjunto, los hallazgos de este capítulo sugieren que la aumentación de datos, cuando se utiliza en combinación con AG, puede ser una herramienta efectiva en la selección de características, especialmente en contextos donde la alta dimensionalidad y escasez muestral, el desbalance de clases y el ruido dificultan la tarea. Asimismo, se destaca que el éxito de esta técnica depende críticamente de la calidad de los datos sintéticos generados y de la adecuada configuración de los modelos subyacentes, lo que subraya la importancia de ajustar las metodologías de manera específica para cada tipo de conjunto de datos.

# Capítulo 5

# Conclusiones y trabajos futuros

A partir de los resultados obtenidos en los experimentos, se torna evidente que la aumentación de datos mediante AV y su combinación con AG es una estrategia prometedora en la selección de características. Para maximizar el potencial de este enfoque, es posible avanzar en varias direcciones, tanto en términos de optimización de los modelos AV y AVC, como en la integración y encadenamiento de modelos más complejos. En este capítulo, repasamos las conclusiones de nuestro trabajo y presentamos los próximos pasos para continuar mejorando los modelos aquí estudiados.

## 5.1. Conclusiones

En esta tesis, abordamos los desafíos que la alta dimensionalidad y escasez muestral, el desbalance de clases y el ruido, plantean a los modelos de aprendizaje automático y, en particular, en los métodos de selección de características basados en Algoritmos Genéticos (AG). Siguiendo los objetivos presentados en el Capítulo 1, propusimos como contribución principal de nuestro trabajo abordar dichos problemas considerando la generación sintética de datos mediante Autocodificadores Variacionales (AV) como estrategia para asistir el proceso de selección de características con AG, con el objetivo de mejorar la capacidad de búsqueda y mitigar los efectos negativos de la falta de datos, el ruido y el desbalance.

Los experimentos descriptos en el Capítulo 3 pusieron en evidencia la eficacia de los AV/AVC para capturar la estructura subyacente de distintos conjuntos de datos, manteniendo la propiedades estadísticas necesarias para que los modelos predictivos entrenados con dichas muestras sintéticas alcanzaran rendimientos comparables —o incluso superiores— a los entrenados con datos reales. Este hallazgo fue especialmente valioso en conjuntos de datos de alta dimensionalidad y pocas observaciones, donde la generación de muestras sintéticas logró resultados positivos en términos de precisión y reducción del impacto negativo del ruido.

Posteriormente, tal como se detalla en el Capítulo 4, integramos este enfoque de aumentación de datos con un AG destinado a la selección de características. En un escenario sin aumentación, el desempeño de los AG puede verse restringido por la falta de datos suficientes para evaluar la calidad de cada subconjunto de variables, agravando el riesgo de convergencia prematura o sobreajuste a muestras escasas. Sin embargo, los experimentos realizados demuestran que la estrategia de aumentar los datos con muestras sintéticas generadas mediante AV/AVC aporta beneficios tanto en el rendimiento final como en la estabilidad de la búsqueda cuando el problema es realmente desafiante.

En efecto, en escenarios con métricas cercanas a saturación (Leukemia y Gisette), donde los modelos sin aumentación ya logran una alta precisión, la generación de datos sintéticos no produjo mejoras estadísticamente significativas en la exactitud. Sin embargo, sí mostró una ventaja en la estabilidad (disminución de la varianza de los resultados) y en la eficiencia de la selección, traducida en subconjuntos de características levemente más compactos.

Sin embargo, en escenarios particularmente complejos o ruidosos (Madelon, GCM y All-Leukemia) el uso de datos aumentados generó mejoras sustanciales. En Madelon, la precisión aumentó más de un 10 %, atribuido a la capacidad de los AV para generar ejemplos sintéticos que contribuye a resaltar las pocas características realmente útiles y compensan la gran proporción de ruido. En GCM, caracterizado por el desbalance de múltiples clases y la alta dimensionalidad, la estrategia de aplicar AV sobre subespacios de características previamente seleccionados por un AG propició un salto del 9 % en la exactitud promedio. Este flujo de trabajo —encadenando selección, generación y nueva selección— resultó clave para gestionar la complejidad y mejorar la representatividad de las muestras sintéticas. La confirmación de esta aproximación vino dada por el dataset All-Leukemia, donde se observó nuevamente que la combinación de AV y AG incrementa la precisión de manera consistente cuando el número de variables se reduce agresivamente, y el problema está lejos de estar "resuelto" por un modelo simple.

Se identificó un umbral más allá del cual continuar generando muestras puede llevar a un sobreajuste del modelo, por ende, perjudicar la clasificación. Así, no se trata solo de una relación lineal donde "más datos" se traduce siempre como "mejores resultados". La pertinencia de la cantidad y diversidad de los ejemplos debe calibrarse cuidadosamente según la complejidad del problema y la calidad de la reconstrucción lograda por el AV.

Por otro lado, a través de la exploración de hiperparámetros, comprobamos que arquitecturas sencillas (por ejemplo, redes de tres capas) o espacios latentes de dimensión moderada pueden ser tan efectivas como configuraciones más complejas o espacios latentes más extensos. En algunos casos, una dimensionalidad excesiva aumenta la redundancia y el ruido, reduciendo la capacidad de generalización de los modelos.

En lo que respecta a consideraciones metodológicas, la experiencias en GCM y All-Leukemia ilustran que el orden en el flujo de trabajo (primero seleccionar un subconjunto de características y luego generar datos sintéticos) puede marcar diferencias importantes en el rendimiento final. Esto sugiere que la sinergia entre selección de características y AV es un campo fértil para investigar configuraciones híbridas y ciclos iterativos adicionales (p.ej., retroalimentar el AV con características cada vez más relevantes).

En conjunto, los resultados presentados confirman la validez de la propuesta: la combinación de Autocodificadores Variacionales y Algoritmos Genéticos puede asistir y mejorar la selección de características, sobre todo en escenarios con alta complejidad, número limitado de muestras y distribuciones desequilibradas. Estos hallazgo cierra nuestra propuesta de investigación mostrando que la generación sintética de datos no solo añade ejemplos a la base de entrenamiento, sino que, al interactuar con el proceso iterativo de un AG, amplía el espacio de exploración efectivo y mejora la confiabilidad en la evaluación de los subconjuntos de variables. De esta forma, la tesis aporta una estrategia integral que une la generación sintética y la selección de características para escenarios de aprendizaje automático complejo.

Los resultados también sugieren que existe un espacio amplio para la creatividad y mejora mediante la optimización de los modelos AV y AVC, la optimización de la integración AV-AG, y la exploración de arquitecturas más avanzadas basadas en encadenamientos y ensambles de modelos. Considerando la complejidad de los dataset reales, quizás los próximos pasos deberían priorizar la construcción de soluciones más robustas y adaptativas, capaces de abordar la complejidad de los problemas con la menor cantidad de presupuestos posibles.

## 5.2. Trabajos futuros

### 5.2.1. Optimización de modelos AV y AVC

Entendemos que uno de los primeros pasos para mejorar la calidad de los datos sintéticos generados pasa por optimizar los modelos AV y AVC. Aunque las versiones creadas en el marco de la presente investigación han mostrado ser efectivas en los contextos de experimentación planteados, como en Madelon y en GCM, es necesario optimizar su capacidad para capturar mejor las estructuras subyacentes de los datos reales. Para lograr esto, se proponen dos líneas de trabajo:

#### 5.2.1.1. Mejora en la función de pérdida

La función de pérdida del AV/AVC desempeña un papel fundamental en la calidad de las muestras generadas. Como vimos en los experimentos de GCM, la divergencia KL puede dominar el proceso de regularización del espacio latente, lo que lleva a una reconstrucción insuficiente de los datos originales. Para mitigar este problema, propondríamos -tal cual lo adelantado en el Capítulo 4- una modificación de la función de pérdida, donde se ajusten los pesos entre la pérdida de reconstrucción y la divergencia KL, dando mayor importancia a la precisión en la reconstrucción. Adicionalmente, se pueden explorar técnicas como la Inferencia Variacional Recocida (Annealed Variational Inference, Huang et al. (2018) ), que ajusta gradualmente el peso de la divergencia KL durante el entrenamiento, permitiendo una transición más suave y efectiva en la regularización del espacio latente.

#### 5.2.1.2. Uso de AV jerárquicos

Otra línea de trabajo posible es la utilización de Autocodificadores Variacionales Jerárquicos (Vahdat and Kautz (2020)). Estos modelos permiten la representación de características en múltiples niveles de abstracción, lo que podría ser particularmente útil para capturar relaciones complejas en conjuntos de datos como GCM. Al incorporar un nivel adicional de complejidad, los HVAEs podrían generar datos sintéticos que no solo preserven mejor la estructura de los datos originales, sino que también mejoren la capacidad del AG para identificar características relevantes en escenarios de alta dimensionalidad.

### 5.2.2. Integración AV-AG optimizada

Aunque los resultados iniciales con la integración AV-AG son alentadores, es posible explorar maneras más eficientes de combinar ambos modelos. El flujo de trabajo encadenado que involucra selección-generación-selección mostró ser efectivo en los experimentos con GCM, pero podría beneficiarse sustancialmente con ciertos ajustes adicionales en su implementación.

#### 5.2.2.1. Selección dinámica de características

Así, una opción es, en lugar de aplicar el AG sobre la totalidad de las características de manera uniforme, podríamos implementar un proceso dinámico de selección de características en múltiples etapas. En este enfoque, el AG se aplicaría inicialmente sobre subconjuntos reducidos de características, optimizando en función de la estabilidad y relevancia de los atributos seleccionados. Finalmente, se combinarían los subconjuntos con mejor desempeño, para formar un dataset de baja dimensiones y alto contenido informativo. Posteriormente, los AV generarían datos sintéticos sólo tomando como inputs este último dataset, reduciendo aún más el ruido y permitiendo una exploración más eficiente del espacio de búsqueda. Todo esto, alineado con la técnica implementada en este trabajo, pero radicalizada en su intención y objetivo.

#### 5.2.2.2. Optimización conjunta de AV y AG

En los experimentos actuales, el AV y el AG se entrenan de manera secuencial y separada, lo que puede limitar la sinergia entre ambos modelos. Un enfoque alternativo sería la optimización conjunta, donde los parámetros de ambos modelos se ajusten simultáneamente durante el entrenamiento. De esta forma, el AV/AVC podría generar muestras sintéticas que maximicen directamente la eficacia del AG en la selección de características, permitiendo una retroalimentación continua entre ambos procesos y una mejora en la calidad del conjunto de datos aumentado.

Reconocemos que, según la experiencia ganada en esta investigación, ajustar los parámetros del AVC y AG, aún de forma independiente, no es un proceso trivial. Particularmente la cantidad de muestras sintéticas en la etapa generativa y el tamaño del cromosoma activo en la etapa de selección resultan parámetros de un impacto crítico en los resultados de la arquitectura.

Pero advertimos también que, en el diseño donde el AV genera muestras sintéticas que luego son utilizadas por el AG para la selección, los modelos no comparten información durante sus respectivas fases de entrenamiento. La optimización conjunta permitiría entrenar ambos modelos de forma simultánea, posibilitando una retroalimentación directa y continua entre los procesos de generación de datos sintéticos (por el AV) y la selección de características (por el AG). Es decir, se ajustaría el proceso de generación de datos en función de la capacidad del AG para seleccionar características relevantes, logrando que el AV genere datos específicamente diseñados para maximizar el rendimiento del AG.

Aunque esta propuesta entaña desafíos inocultables (diseño de una función de pérdida, coordinación entre los procesos de optimización, capacidad computacional, por mencionar algunas), también ofrecería beneficios de gran valor. En particular, podríamos disponer de una mayor sinergia entre generación y selección. Esto se produciría a través de una retroalimentación directa entre los modelos, haciendo que el AV se adapte mejor a las necesidades del AG, generando datos que aborden mejor los desafíos específicos de selección de características en cada problema. Al entrenar el AV para generar datos que optimicen el desempeño del AG, el proceso de búsqueda de características relevantes podría volverse más eficiente. El AG podría explorar de manera más efectiva el espacio de características, identificando subconjuntos más precisos y reduciendo la dimensionalidad sin perder información relevante.

### 5.2.3. Exploración de encadenamientos y stacks de modelos

Los resultados obtenidos sugieren que una única iteración del proceso AV-AG puede no ser suficiente para capturar completamente las relaciones entre características en problemas altamente complejos. En este contexto, la construcción de arquitecturas más complejas mediante el encadenamiento de varios modelos (stacks) o la integración de ensambles, similar a los Bosques Aleatorios, se presenta como una vía interesante de investigación.

Una posibilidad es el diseño de un stack de AV y AG, donde varias instancias de ambos modelos se encadenen de manera secuencial o paralela. En este esquema, por ejemplo, una primer secuencia AG-AV generaría un conjunto de datos sintéticos, que luego alimentaría a una segunda secuencia de AG-AG con configuraciones más específicas. Este proceso de encadenamiento podría permitir una mayor concentración de la generación y selección, especialmente en conjuntos de datos donde las relaciones entre las características son extremadamente complejas.

Al igual que los Bosques Aleatorios combinan múltiples árboles de decisión para mejorar la precisión y la robustez del modelo, se puede explorar la creación de un ensamble de AV y AG. Este enfoque involucraría el entrenamiento de múltiples AV y AG con diferentes configuraciones y subconjuntos de datos, cuyas salidas se combinarían para producir una solución más robusta. Los ensambles suelen ser efectivos para reducir la varianza de los modelos individuales, lo que podría resultar en una selección de características más estable y en un rendimiento más consistente.

### 5.2.4. Exploración de arquitecturas híbridas y meta-aprendizaje

Por último, se abre la posibilidad de explorar arquitecturas híbridas que combinen AV y AG con otros enfoques de aprendizaje automático, como los algoritmos de meta-aprendizaje. Estos modelos podrían ser entrenados para aprender a seleccionar automáticamente los mejores hiperparámetros y configuraciones para cada conjunto de datos, adaptándose dinámicamente a las características específicas del problema.

En lugar de fijar los parámetros del AG a priori, el meta-aprendizaje permitiría que el AG aprenda automáticamente cuáles son los mejores parámetros en función de la estructura de los datos. Este enfoque podría incluir la selección adaptativa del tamaño del cromosoma activo, las tasas de mutación y cruce, y el número de generaciones, optimizando el rendimiento del AG en cada iteración.

# Bibliografía

Ai, Qingzhong, Pengyun Wang, Lirong He, Liangjian Wen, Lujia Pan, and Zenglin Xu. 2023. "Generative Oversampling for Imbalanced Data via Majority-Guided VAE." February 14, 2023. http://arxiv.org/abs/2302.10910.

Almugren, Nada, and Hala Alshamlan. 2019. "A Survey on Hybrid Feature Selection Methods in Microarray Gene Expression Data for Cancer Classification." IEEE Access 7: 78533–48. https://doi.org/10.1109/ACCESS.2019.2922987.

Blaauw, Merlijn, and Jordi Bonada. 2016. "Modeling and Transforming Speech Using Variational Autoencoders." In Interspeech 2016, 1770–74. ISCA. https://doi.org/10.21437/Interspeech.2016-1183.

Blagus, Rok, and Lara Lusa. 2013. "SMOTE for High-Dimensional Class-Imbalanced Data." BMC Bioinformatics 14 (1): 106. https://doi.org/10.1186/1471-2105-14-106.

Bolón-Canedo, Verónica, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. 2015. Feature Selection for High-Dimensional Data. Artificial Intelligence: Foundations, Theory, and Algorithms. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-21858-8.

Boom, Cedric De, Samuel Wauthier, Tim Verbelen, and Bart Dhoedt. 2021. "Dynamic Narrowing of VAE Bottlenecks Using GECO and L0 Regularization." April 13, 2021. https://doi.org/10.48550/arXiv.2003.10901.

Dang, Hien, Tho Tran, Tan Nguyen, and Nhat Ho. 2024. "Beyond Vanilla Variational Autoencoders: Detecting Posterior Collapse in Conditional and Hierarchical Variational Autoencoders." May 13, 2024. https://doi.org/10.48550/arXiv.2306.05023.

Doersch, Carl. 2021. "Tutorial on Variational Autoencoders." January 3, 2021. http://arxiv.org/abs/1606.05908.

El-Hasnony, Ibrahim M., Sherif I. Barakat, Mohamed Elhoseny, and Reham R. Mostafa. 2020. "Improved Feature Selection Model for Big Data Analytics." IEEE Access 8: 66989–7004. https://doi.org/10.1109/ACCESS.2020.2986232.

Fajardo, Val Andrei, David Findlay, Charu Jaiswal, Xinshang Yin, Roshanak Houmanfar, Honglei Xie, Jiaxi Liang, Xichen She, and D. B. Emerson. 2021. "On Oversampling Imbalanced Data with Deep Conditional Generative Models." Expert Systems with Applications 169 (May): 114463. https://doi.org/10.1016/j.eswa.2020.114463.

Fisher, Ronald. 1935. The Design of Experiments. London: Oliver and Boyd.

Goldberg, David E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. New York, NY, USA: Addison-Wesley.

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, et al. 1999. "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring." Science 286 (5439): 531–37. https://doi.org/10.1126/science.286.5439.531.

Hastie, T, R Tibshirani, and J Friedman. 2009. The Element of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition. Springer.

Huang, Chin-Wei, Shawn Tan, Alexandre Lacoste, and Aaron Courville. 2018. "Improving Explorability in Variational Inference with Annealed Variational Objectives."

arXiv.org. September 6, 2018. https://arxiv.org/abs/1809.01818v3.

Isabelle Guyon, Steve Gunn. 2004. "Gisette." UCI Machine Learning Repository. https://doi.org/10.24432/C5HP5B.

Jiao, Ruwang, Bach Hoai Nguyen, Bing Xue, and Mengjie Zhang. 2023. "A Survey on Evolutionary Multiobjective Feature Selection in Classification: Approaches, Applications, and Challenges." IEEE Transactions on Evolutionary Computation, 1–1. https://doi.org/10.1109/TEVC.2023.3292527.

Khmaissia, Fadoua, and Hichem Frigui. 2023. "Confidence-Guided Data Augmentation for Improved Semi-Supervised Training." February 21, 2023. http://arxiv.org/abs/2209.08174.

Kingma, Diederik P., and Max Welling. 2019. "An Introduction to Variational Autoencoders." Foundations and Trends® in Machine Learning 12 (4): 307–92. https://doi.org/10.1561/2200000056.

Kwarciak, Kamil, and Marek Wodzinski. 2023. "Deep Generative Networks for Heterogeneous Augmentation of Cranial Defects." August 9, 2023. http://arxiv.org/abs/2308.04883.

Latif, Siddique, Rajib Rana, Junaid Qadir, and Julien Epps. 2020. "Variational Autoencoders for Learning Latent Representations of Speech Emotion: A Preliminary Study." July 27, 2020. https://doi.org/10.48550/arXiv.1712.08708.

Leelarathna, Navindu, Andrei Margeloiu, Mateja Jamnik, and Nikola Simidjievski. 2023. "Enhancing Representation Learning on High-Dimensional, Small-Size Tabular Data: A Divide and Conquer Method with Ensembled VAEs." June 27, 2023. http://arxiv.org/abs/2306.15661.

Li, Chenghao, and Chaoning Zhang. 2023. "When ChatGPT for Computer Vision Will Come? From 2d to 3d." 2023. https://doi.org/10.48550/ARXIV.2305.06133.

Liu, Qi, Miltiadis Allamanis, Marc Brockschmidt, and Alexander Gaunt. 2018. "Constrained Graph Variational Autoencoders for Molecule Design." In Advances in Neural Information Processing Systems. Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/hash/b8a03c5c15fcfa8dae0b03351eb1742f-Abstract.html.

Ramaswamy, Sridhar, Pablo Tamayo, Ryan Rifkin, Sayan Mukherjee, Chen-Hsiang Yeang, Michael Angelo, Christine Ladd, et al. 2001. "Multiclass Cancer Diagnosis Using Tumor Gene Expression Signatures." Proceedings of the National Academy of Sciences 98 (26): 15149–54. https://doi.org/10.1073/pnas.211566398.

Ramchandran, Siddharth, Gleb Tikhonov, Otto Lönnroth, Pekka Tiikkainen, and Harri Lähdesmäki. 2022. "Learning Conditional Variational Autoencoders with Missing Covariates." March 2, 2022. http://arxiv.org/abs/2203.01218.

Roberts, Adam, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2019. "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music." November 11, 2019. http://arxiv.org/abs/1803.05428.

Solorio-Fernández, Saúl, J. Ariel Carrasco-Ochoa, and José Fco. Martínez-Trinidad. 2020. "A Review of Unsupervised Feature Selection Methods." Artificial Intelligence Review 53 (2): 907–48. https://doi.org/10.1007/s10462-019-09682-y.

Torre, Jordi de la. 2023. "Autocodificadores Variacionales (VAE) Fundamentos Te\'oricos y Aplicaciones." February 18, 2023. https://doi.org/10.48550/arXiv.2302.09363.

Vahdat, Arash, and Jan Kautz. 2020. "NVAE: A Deep Hierarchical Variational Autoencoder." In Advances in Neural Information Processing Systems, 33:19667–79. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/hash/e3b21256183cf7c2c7a66be163579d37-Abstract.html.

Vie, Aymeric, Alissa M. Kleinnijenhuis, and Doyne J. Farmer. 2021. "Qualities, Challenges and Future of Genetic Algorithms: A Literature Review." September 13, 2021. http://arxiv.org/abs/2011.05277.

Vignolo, Leandro D., and Matias F. Gerard. 2017. "Evolutionary Local Improvement on Genetic Algorithms for Feature Selection." In 2017 XLIII Latin American Computer Conference (CLEI), 1–8. Cordoba: IEEE. https://doi.org/10.1109/CLEI.2017.8226467.

Zhang, Rui, Feiping Nie, Xuelong Li, and Xian Wei. 2019. "Feature Selection with Multi-View Data: A Survey." Information Fusion 50 (October): 158–67. https://doi.org/10.1016/j.inffus.2018.11.019.

Zhang, Yuchi, Yongliang Wang, Liping Zhang, Zhiqiang Zhang, and Kun Gai. 2019. "Improve Diverse Text Generation by Self Labeling Conditional Variational Auto Encoder." March 26, 2019. https://doi.org/10.48550/arXiv.1903.10842.

Apéndice A

# Implementación de un Algoritmo Genético

En el siguiente fragmento de código presentamos las operaciones elementales de un AG, que como dice Goldberg son extraordinariamente sencillas. La secuencia de operaciones se inicializa con un conjunto de soluciones iniciales, denominado población. El ciclo iterativo principal del Algoritmo Genético genera nuevas soluciones candidatas descendientes mediante cruce y mutación hasta que la población esté completa. En cada iteración, los individuos son evaluados mediante una función de aptitud que mide su calidad en relación al problema a resolver (aquí, la función de aptitud es simplemente la suma de los valores de los genes, en contextos reales, esta función se ajusta a las necesidades del problema pudiendo ser una función de costo, una métrica de desempeño, entre otras). Los individuos más aptos son seleccionados para reproducirse, lo que implica la aplicación de operadores genéticos para generar nuevos individuos. Este proceso se repite a lo largo de múltiples generaciones, permitiendo que la población evolucione y se adapte a las condiciones del problema.

## A.1. Algoritmo Genético básico en python

```
import random

# Parámetros del Algoritmo Genético
num_individuals = 5
chromosome_length = 10
num_generations = 10
mutation_rate = 0.1

# Inicializar la población con individuos aleatorios
individuals = [random.randint(0, 1) for _ in range(chromosome_length)]
population = [individuals for _ in range(num_individuals)]

# Ejecutar el Algoritmo Genético
for generation in range(num_generations):
    # Calcular aptitud
    fitness_values = [sum(ind) for ind in population]

    # Crear nueva población
    new_population = []
```

```
    while len(new_population) < num_individuals:
        # Selección de dos padres
        parent1 = random.choices(population, weights=fitness_values)[0]
        parent2 = random.choices(population, weights=fitness_values)[0]

        # Cruce de un punto
        point = random.randint(1, chromosome_length - 1)
        child1 = parent1[:point] + parent2[point:]
        child2 = parent2[:point] + parent1[point:]

        # Mutación
        for i in range(chromosome_length):
            if random.random() < mutation_rate:
                child1[i] = 1 - child1[i]
            if random.random() < mutation_rate:
                child2[i] = 1 - child2[i]

        new_population.append(child1)
        if len(new_population) < num_individuals:
            new_population.append(child2)

    # Reemplazar la población antigua con la nueva
    population = new_population

    # Mostrar la población actual y sus aptitudes
    print(f"Generación {generation + 1}:")
    for ind in population:
        print(f"Individuo: {ind}, Aptitud: {sum(ind)}")
    print()
```

En el caso de nuestro trabajo, se implementó un AG con la librería `DEAP` de Python, que puede ser adaptado para la selección de características en diferentes contextos.

## A.2. Algoritmo Genético con la librería DEAP

```
import random
import numpy as np
from deap import base, creator, tools
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score

# Parámetros del Algoritmo Genético
PROB_MUT = 0.1        # Probabilidad de mutación
PX = 0.75             # Probabilidad de cruce
GMAX = 15             # Número máximo de generaciones
POP_SIZE = 20         # Tamaño de la población
```

```
# Carga del conjunto de datos de ejemplo
data = load_breast_cancer()
X = data.data
y = data.target

# División del conjunto de datos en entrenamiento y prueba
Xtrain, Xtest, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.3,
    random_state=42
)

# Tamaños derivados
DAT_SIZE = Xtrain.shape[0]
IND_SIZE = Xtrain.shape[1]
PM = 1 / IND_SIZE    # Probabilidad de mutación por gen

# Definición de la función de fitness
def fitness(individual, Xtrain, Xtest, y_train, y_test):
    """Función de aptitud para evaluar la calidad de un individuo."""
    if not any(individual):
        return 0,  # Evita seleccionar individuos sin genes activos

    X_train = Xtrain[:, individual]
    X_test = Xtest[:, individual]

    model = MLPClassifier(hidden_layer_sizes=(5, 3),
        max_iter=1000,
        random_state=42
    )
    model.fit(X_train, y_train)

    predictions = model.predict(X_test)
    accuracy = accuracy_score(y_test, predictions)

    # Minimización del número de características seleccionadas
    n_genes = np.sum(individual)
    alpha = 0.5  # Ponderación entre precisión y número de genes

    return alpha * accuracy + (1 - alpha) * (1 - n_genes / IND_SIZE),

# Configuración del entorno evolutivo
creator.create("FitnessMax", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMax)

toolbox = base.Toolbox()
toolbox.register("attr_bool", lambda: random.random() < PM)
toolbox.register("individual", tools.initRepeat,
                    creator.Individual, toolbox.attr_bool, n=IND_SIZE)
```

```
toolbox.register("population", tools.initRepeat, list, toolbox.individual)

toolbox.register("mate", tools.cxTwoPoint)
toolbox.register("mutate", tools.mutFlipBit, indpb=PROB_MUT)
toolbox.register("select", tools.selTournament, tournsize=3)
toolbox.register("evaluate", fitness,
                    Xtrain=Xtrain,
                    Xtest=Xtest,
                    y_train=y_train,
                    y_test=y_test)

# Función principal del Algoritmo Genético
def main():
    # Inicialización de la población
    population = toolbox.population(n=POP_SIZE)

    # Evaluación inicial
    fitnesses = list(map(toolbox.evaluate, population))
    for ind, fit in zip(population, fitnesses):
        ind.fitness.values = fit

    # Bucle evolutivo
    for gen in range(GMAX):
        # Selección y reproducción
        offspring = toolbox.select(population, len(population))
        offspring = list(map(toolbox.clone, offspring))

        # Aplicación del cruce y mutación
        for child1, child2 in zip(offspring[::2], offspring[1::2]):
            if random.random() < PX:
                toolbox.mate(child1, child2)
                del child1.fitness.values
                del child2.fitness.values

        for mutant in offspring:
            if random.random() < PROB_MUT:
                toolbox.mutate(mutant)
                del mutant.fitness.values

        # Evaluación de los nuevos individuos
        invalid_ind = [ind for ind in offspring if not ind.fitness.valid]
        fitnesses = map(toolbox.evaluate, invalid_ind)
        for ind, fit in zip(invalid_ind, fitnesses):
            ind.fitness.values = fit

        # Reemplazo de la población
        population[:] = offspring

        # Recopilación de estadísticas
```

```
        fits = [ind.fitness.values[0] for ind in population]
        print(f"Gen:{gen + 1}-Mejor_fit:{max(fits):.4f}-Promedio:{np.mean(fits):.4f}")

    # Mejor individuo al finalizar
    best_ind = tools.selBest(population, 1)[0]
    print("\nMejor individuo encontrado: ", best_ind)
    print(f"Fitness: {best_ind.fitness.values[0]:.4f}")
    print(f"Número de características seleccionadas: {np.sum(best_ind)}")

if __name__ == "__main__":
    main()
```

Apéndice B

# Teorema de esquemas para AG

El teorema de esquemas (Goldberg, David E. 1989) predice el número esperado de copias de un esquema $H$ en la próxima generación $t+1$, dado su número de copias en la generación actual $t$. Se expresa de la siguiente manera:

$m(H,t+1) \geq m(H,t) \cdot \frac{f(H)}{\bar{f}} \cdot \left[1 - p_c \frac{\delta(H)}{l-1}\right] \cdot (1-p_m)^{o(H)}$

Donde: - $m(H,t)$ es el número de copias del esquema $H$ en la generación $t$. - $f(H)$ es la aptitud promedio de los individuos que pertenecen al esquema $H$. - $\bar{f}$ es la aptitud promedio de la población total. - $p_c$ es la probabilidad de cruce. - $\delta(H)$ es la longitud de definición del esquema $H$, que es la distancia entre el primer y el último gen fijo en el esquema. - $l$ es la longitud del cromosoma. - $p_m$ es la tasa de mutación. - $o(H)$ es el orden del esquema, es decir, el número de posiciones fijas en el esquema.

Consideremos un ejemplo con los siguientes parámetros:

- Longitud del cromosoma: $l = 6$
- Esquema $H = 1*0*1*$ (donde $*$ puede ser 0 o 1)
- Población actual tiene 100 individuos.
- $m(H,t) = 20$ (es decir, 20 individuos coinciden con el esquema $H$).
- Aptitud promedio de la población $\bar{f} = 15$.
- Aptitud promedio de los individuos que coinciden con el esquema $H$, $f(H) = 18$.
- Probabilidad de cruce $p_c = 0.7$.
- Tasa de mutación $p_m = 0.01$.
- Longitud de definición del esquema $\delta(H) = 4$ (dado que las posiciones fijas son 1, 3 y 5, la distancia entre las posiciones es 4).
- Orden del esquema $o(H) = 3$ (el número de posiciones fijas es 3).

Aplicando estos valores al teorema del esquema:

1. Factor de Selección:

   $\frac{f(H)}{\bar{f}} = \frac{18}{15} = 1.2$

   Esto indica que los individuos que coinciden con el esquema $H$ tienen una aptitud superior a la media y, por lo tanto, es más probable que sean seleccionados.

2. Probabilidad de Conservación ante el Cruce:

   $1 - p_c \frac{\delta(H)}{l-1} = 1 - 0.7 \cdot \frac{4}{6-1} = 1 - 0.7 \cdot 0.8 = 1 - 0.56 = 0.44$

   Hay un 44% de probabilidad de que el esquema $H$ se conserve tras el cruce.

3. Probabilidad de Conservación ante la Mutación:

   $(1-p_m)^{o(H)} = (1-0.01)^3 = 0.99^3 \approx 0.9703$

El esquema $H$ tiene aproximadamente un 97% de probabilidad de no ser destruido por la mutación.

4. Cálculo Final:

   $m(H,t+1) \geq 20 \cdot 1.2 \cdot 0.44 \cdot 0.9703 \approx 20 \cdot 0.5127 = 10.254$

   Por lo tanto, en la próxima generación, se espera que haya al menos 10 copias del esquema $H$ en la población.

Este cálculo muestra cómo el esquema $H$, que tiene una aptitud superior a la media y ciertas características de proximidad posicional (es decir, una longitud de definición baja), es favorecido en la reproducción y es probable que se mantenga en la población.