

# Detección del cáncer mediante modelos de clasificación de miARNs



viu

Universidad  
Internacional  
de Valencia

**Trabajo de Fin de Máster**

**Titulación:**

Máster Universitario en Big  
Data y Ciencia de Datos

**Curso académico:**

2023 – 2024

**Alumno/a:**

Nuñez Martinez, Malena

**D.N.I:**

79051602G

**Director/a de TFM:**

Lebron Aguilar, Ricardo

**Convocatoria:**

Primera

**Fecha:**

Octubre-2024

De:

 Planeta Formación y Universidades

# Índice

Resumen .....	8
Abstract .....	9
1. Introducción .....	10
2. Objetivos.....	13
3. Marco teórico.....	14
3.1. El cáncer .....	14
3.2. Biología molecular de la célula .....	14
3.3. miARN (microARN) .....	15
3.4. Biopsias líquidas .....	15
3.5. Tecnología RNA-seq y análisis de datos de miARNs .....	15
3.6. Técnicas de normalización .....	16
3.7. Modelos de clasificación aplicados a datos biomédicos .....	18
3.8. Evaluación de modelos .....	20
3.9. Fuga de datos ( <i>data leakage</i> ).....	21
4. Estado de Arte .....	22
4.1. MiARNs como biomarcadores en el cáncer .....	22
4.2. Biopsias líquidas en la detección del cáncer .....	22
4.3. Tecnologías de secuenciación y análisis de miARNs .....	23
4.4. Modelos de clasificación aplicados a la detección de cáncer.....	23
4.5. Técnicas de procesamiento y normalización de datos RNA-seq .....	24
4.6. Selección de biomarcadores de miARN .....	25
4.7. Estudios comparativos .....	25
5. Metodología.....	26
5.1. Carga de datos.....	26
5.2. Análisis exploratorio de datos inicial .....	27
5.2.1. Exploración de miARNs .....	27
5.2.2. Exploración de la metadata .....	30
5.3. Preprocesamiento .....	30
5.3.1. Valores duplicados .....	30
5.3.2. Filtrado de miARNs con baja expresión .....	30
5.3.3. Normalización de miARNs .....	31

5.3.4.	Estandarización de datos.....	31
5.3.5.	Integración de datos y división en conjuntos de entrenamiento y prueba 31	
5.3.6.	Valores faltantes en <i>metadata</i> .....	32
5.3.7.	Codificación de variables categóricas en metadata .....	33
5.3.8.	Valores atípicos en miARN .....	33
5.3.9.	Valores atípicos en miARN y metadata.....	34
5.3.10.	Imputación de valores faltantes en metadata.....	35
5.3.11.	Balanceo de la clase objetivo <i>Cancer</i> .....	36
5.4.	Análisis exploratorio de datos posprocesamiento .....	36
5.5.	Selección de miARNs.....	37
5.6.	Construcción de modelos de clasificación .....	39
5.7.	Evaluación de modelos .....	40
5.8.	Importancia de miARNs.....	41
6.	Resultados y discusión .....	42
6.1.	Análisis exploratorio de miARNs.....	42
6.2.	Análisis exploratorio de <i>metadata</i> .....	49
6.3.	Preprocesamiento de miARNs y <i>metadata</i> .....	50
6.3.1.	Filtrado de miARNs con baja expresión .....	50
6.3.2.	Normalización de datos mediante TMM .....	51
6.3.3.	Estandarización de datos con $\log_2(x + 1)$ .....	52
6.3.4.	Tratamiento de valores faltantes en <i>metadata</i> .....	53
6.3.5.	Análisis de valores atípicos.....	54
6.3.6.	Balanceo de clases.....	57
6.4.	Análisis exploratorio de datos posprocesamiento .....	57
6.5.	Evaluación de los miARNs seleccionados .....	62
6.6.	Optimización de los modelos de clasificación .....	65
6.7.	Elección del mejor modelo de clasificación.....	66
6.8.	Importancia de miARNs.....	68
7.	Conclusiones y Trabajo futuro .....	70
	Referencias .....	72
	Apéndice I: Estadísticas descriptivas .....	81
	Apéndice II: Detección de valores atípicos.....	82
	<i>miARN</i> .....	82



<i>miARN y metadata</i> .....	87
Apéndice III: Curvas de aprendizaje .....	89
Apéndice IV: Evaluación de modelos .....	97

# Índice de ilustraciones

Figura 1: Número de casos por cáncer. Nota: Adaptado de Cancer Today   IARC (2024).	10
Figura 2: Supervivencia en 5 años por grados de cáncer de colon. Nota: Adaptado de ICBP - SURVMARK2 (s. f.).	11
Figura 3: MI - Curva de validación para seleccionar el número de características.	38
Figura 4: SVM + L1 - Curva de validación para seleccionar el valor del hiperparámetro C.	38
Figura 5: RF - Curva de validación para seleccionar el número de características.	39
Figura 6: Histogramas antes de la normalización.	43
Figura 7: Histogramas después de la normalización.	44
Figura 8: Diagramas de caja de diferentes experimentos.	45
Figura 9: Mapa de calor de los 25 miARNs con mayor correlación absoluta.	46
Figura 10: Top 40 miARNs correlacionados con la variable cáncer.	46
Figura 11: Análisis PCA de los experimentos.	47
Figura 12: Healthy y Cancer - Gráfico de dispersión.	48
Figura 13: Fluid - Gráfico de dispersión.	48
Figura 14: Análisis de clustering.	49
Figura 15: Análisis PCA de miARNs.	49
Figura 16: Distribución de las categorías de la metadata.	50
Figura 17: Histogramas después de la normalización con TMM.	52
Figura 18: Histogramas después de la normalización con $\log_{2}x + 1$ .	53
Figura 19: Mapa de calor para verificar valores faltantes. El gráfico de la derecha corresponde al conjunto de datos de entrenamiento y el de la izquierda al de prueba.	53
Figura 20: Valores atípicos en miARN. Los puntos naranjas corresponden a muestras de cáncer.	55
Figura 21: Valores atípicos en miARNs. Los puntos rojos corresponden a las muestras eliminadas.	56
Figura 22: Valores atípicos en miARNs junto con metadata. Los puntos rojos corresponden a las muestras eliminadas.	57
Figura 23: Histogramas tras el preprocesamiento de datos.	59
Figura 24: Diagramas de caja tras el preprocesamiento de datos.	59
Figura 25: Mapa de calor con correlaciones entre miARNs.	60
Figura 26: Top 50 correlaciones entre miARNs y Cáncer.	61
Figura 27: Cáncer - Gráfico de dispersión.	61
Figura 28: Curva de aprendizaje para evaluar el rendimiento de la selección de miARNs.	63
Figura 29: Mapa de calor con todas las correlaciones entre miARNs tras la selección de características.	64
Figura 30: IF - Detección de valores atípicos.	83
Figura 31: LOF - Detección de valores atípicos.	85

Figura 32: DBSCAN - Detección de valores atípicos.....	<b>¡Error! Marcador no definido.</b>
Figura 33: Mahalanobis - Detección de valores atípicos. ....	87
Figura 34: metadata - IF - Detección de valores atípicos. ....	87
Figura 35: metadata - LOF - Detección de valores atípicos.....	88
Figura 36: SVM - Curvas de aprendizaje - Modelo por defecto, modelo con optimización de hiperparámetros y modelo personalizado. ....	90
Figura 37: KNN - Curvas de aprendizaje - Modelo por defecto, modelo con optimización de hiperparámetros y modelo personalizado. ....	91
Figura 38: XGBoost - Curvas de aprendizaje - Modelo por defecto, Modelo por defecto regularizado, modelo con optimización de hiperparámetros, modelo personalizado 1 y modelo personalizado 2.....	93
Figura 39: Random Forest - Curvas de aprendizaje - Modelo por defecto, modelo con optimización de hiperparámetros y modelo personalizado. ....	95
Figura 40: MLP - Curvas de aprendizaje - Modelo por defecto, modelo con optimización de hiperparámetros.....	96
Figura 41: SVM - Matriz de confusión, curva ROC y curva PR. ....	97
Figura 42: KNN - Matriz de confusión, curva ROC y curva PR.....	98
Figura 43: Random Forest: KNN - Matriz de confusión, curva ROC y curva PR. ....	99
Figura 44: XGBoost: KNN - Matriz de confusión, curva ROC y curva PR.....	99
Figura 45: MLP - Matriz de confusión, curva ROC y curva PR.....	100

# Índice de tablas

<b>Tabla 1:</b> Valores faltantes en mirna y metadata.....	53
<b>Tabla 2:</b> Número de miARNs seleccionados por método.....	62
<b>Tabla 3:</b> Mejores hiperparámetros por modelo. ....	66
<b>Tabla 4:</b> Métricas de evaluación por modelo. ....	67
<b>Tabla 5:</b> metadata - Estadísticas descriptivas antes del procesamiento. ....	81
<b>Tabla 6:</b> metadata - Estadísticas descriptivas antes del procesamiento. ....	81
<b>Tabla 7:</b> metadata - Estadísticas descriptivas antes del procesamiento. ....	81

## Resumen

Este proyecto tiene como objetivo la detección del cáncer mediante la aplicación de modelos de clasificación sobre datos de microARN (miARN) obtenidos de biopsias líquidas. En la primera fase, se llevó a cabo un análisis exploratorio de datos para garantizar la calidad y el estado de los datos antes de la selección de características. Se revisaron los tipos de datos, se realizó un resumen estadístico, se inspeccionaron valores faltantes y atípicos, y se analizaron los diferentes patrones formados por los datos. Los histogramas y gráficos de densidad mostraron una distribución sesgada; sin embargo, se observó una mejora en la asimetría tras el preprocesamiento y la normalización.

A continuación, se aplicó una selección de características que resultó en la identificación de 145 miARNs relevantes. La evaluación de estos miARNs mostró un buen rendimiento en métricas clave, como la AUC-ROC y el F1-Score, con una destacada consistencia en la precisión y la sensibilidad. Posteriormente, se optimizaron cinco modelos de clasificación: SVM, KNN, XGBoost, Random Forest y MLP. Los resultados indicaron que XGBoost fue el modelo más eficaz, alcanzando una sensibilidad del 98,99% y un F1-Score de 0,9159, lo que lo hace especialmente adecuado para la detección de cáncer en este contexto clínico.

Por último, se realizó un análisis de la importancia de los miARNs seleccionados, destacando aquellos que resultaron ser más influyentes en la clasificación. Los hallazgos sugieren que los miARNs hsa-miR-185-5p, hsa-miR-378g y hsa-let-7b-5p son de especial relevancia.

Este trabajo resalta el papel de los miARNs como biomarcadores en la detección temprana del cáncer, lo que abre la puerta a futuras investigaciones y aplicaciones clínicas en la oncología.

Palabras clave:

miARNs, detección del cáncer, modelos de clasificación, biomarcadores, selección de características.

## Abstract

This project aims to detect cancer by applying classification models to microRNA (miRNA) data obtained from liquid biopsies. In the first phase, an exploratory data analysis was performed to ensure data quality and integrity before feature selection. Data types were reviewed, statistical summaries were generated, missing and outlier values were inspected, and different patterns in the data were analyzed. Histograms and density plots revealed skewed distributions; however, an improvement in asymmetry was seen after preprocessing and normalization.

Next, a feature selection process found 145 relevant miRNAs. The evaluation of these miRNAs proved robust performance across key metrics, such as AUC-ROC and F1-Score, showing notable consistency in both precision and sensitivity. Five classification models were then optimized: SVM, KNN, XGBoost, Random Forest, and MLP. Results showed that XGBoost was the most effective model, achieving a sensitivity of 98.99% and an F1-Score of 0.9159, making it particularly suitable for cancer detection in this clinical setting.

Finally, an analysis of the selected miRNAs highlighted those that were most influential in the classification process. The findings suggest that miRNAs hsa-miR-185-5p, hsa-miR-378g, and hsa-let-7b-5p are of special relevance.

This work emphasizes the role of miRNAs as biomarkers in the early detection of cancer, paving the way for future research and clinical applications in oncology.

Keywords:

miRNAs, cancer detection, classification models, biomarkers, feature selection.

# 1. Introducción

El cáncer se ha convertido en uno de los mayores desafíos de salud pública a nivel mundial y es la principal causa de muerte en el mundo. En 2022 hubo 20 millones de nuevos casos de cáncer y 10 millones de fallecimientos (Bray Bsc et al., 2024), siendo los canceres de pulmón, mama y colon los más letales (Figura 1). Según las proyecciones de la Organización Mundial de la Salud, se espera que para el año 2040 haya aproximadamente 29.9 millones de nuevos casos de esta enfermedad, un aumento del 49,5% (Cancer Tomorrow | IARC, 2024). Este panorama resalta la necesidad de mejorar los métodos de diagnóstico y tratamiento del cáncer, así como la importancia de la investigación en esta área.

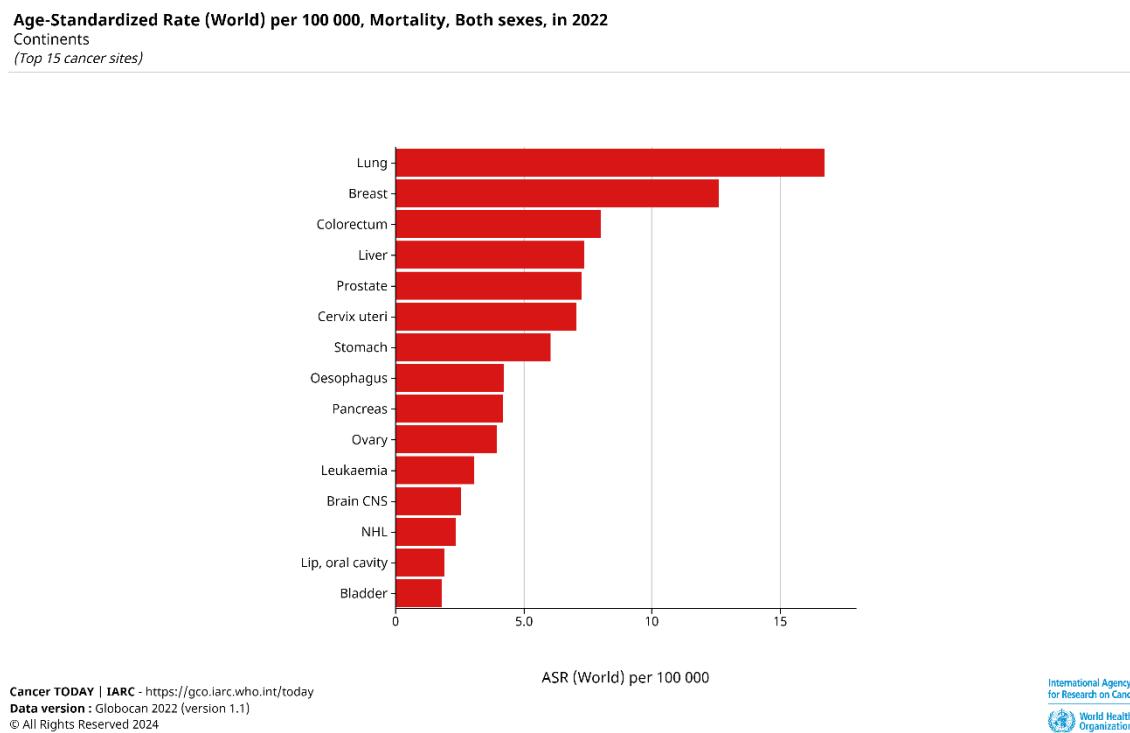


Figura 1: Número de casos por cáncer. Nota: Adaptado de Cancer Today / IARC (2024).

Uno de los aspectos más críticos es la detección temprana del cáncer, que puede influir significativamente en el pronóstico y la eficacia de los tratamientos. La detección en etapas iniciales permite implementar tratamientos más efectivos, lo que a su vez aumenta las tasas de supervivencia (Figura 2). Además, es fundamental acertar con el diagnóstico, ya que cada tipo de cáncer demanda una terapia específica. Entre los tratamientos utilizados se encuentran las cirugías, la radioterapia y las terapias sistémicas, como la quimioterapia, los tratamientos hormonales y las terapias biológicas dirigidas (Bray Bsc et al., 2024). Sin embargo, los métodos de detección actuales, como las biopsias de tejido y las técnicas de imagen, presentan limitaciones. Las biopsias de tejido son invasivas, a menudo requieren procedimientos quirúrgicos y no siempre capturan la heterogeneidad tumoral (Aaltonen et al., 2017). Las técnicas de imagen,

aunque útiles, pueden tener limitaciones en la resolución y en la capacidad para detectar lesiones pequeñas o en localizaciones complicadas (American Cancer Society, 2024).

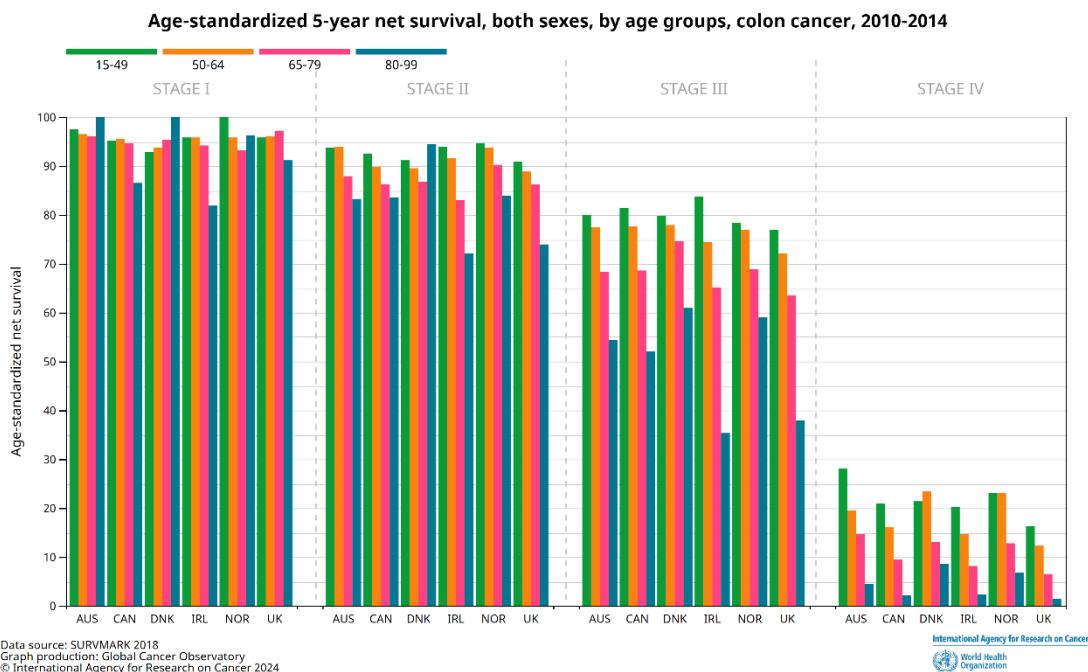


Figura 2: Supervivencia en 5 años por grados de cáncer de colon. Nota: Adaptado de ICBP - SURVMARK2 (s. f.).

En este sentido, los miARNs han emergido como moléculas de gran interés en la investigación del cáncer. Los miARNs son pequeñas moléculas de ácido ribonucleico (ARN) no codificantes que regulan la expresión génica a través de la degradación del ARN mensajero o la inhibición de su traducción. La desregulación de los miARNs ha sido asociada con diversos tipos de cáncer, lo que sugiere su potencial como biomarcadores para la detección y el pronóstico de la enfermedad. Estos biomarcadores no solo permiten una evaluación más precisa del estado del tumor y una obtención menos invasiva de la muestra, sino que también pueden ofrecer información sobre la respuesta al tratamiento y el riesgo de recaída (X. Chen et al., 2008; Croce, 2009; Mitchell et al., 2008; Yang et al., 2017).

Por ende, las biopsias líquidas han surgido como una solución innovadora. Estas técnicas recolectan muestras de fluidos biológicos, como sangre o saliva, lo que las convierte en métodos menos invasivos. Las biopsias líquidas facilitan la detección de miARNs en sangre, ofreciendo la posibilidad de monitorear la enfermedad de manera más frecuente y menos arriesgada. Por lo tanto, este enfoque mejora la capacidad para detectar el cáncer en etapas tempranas (Sestini et al., 2015).

La aplicación de técnicas de ciencia de datos y *machine learning* (ML) en la detección de cáncer representa un gran avance en la investigación biomédica. Mediante el uso de modelos de clasificación, es posible analizar grandes volúmenes de datos de expresión de miARN para identificar patrones complejos que puedan estar relacionados con la

presencia de cáncer. Estas técnicas no solo permiten la identificación de biomarcadores potenciales, sino que también ayudan a predecir la evolución de la enfermedad y la respuesta al tratamiento (Alharbi & Vakanski, 2023; L. Chen et al., 2019; Sarkar et al., 2021).

Los datos utilizados en este trabajo provienen de ***liqDB***, una base de datos diseñada para almacenar perfiles de secuenciación de pequeños ARN (*small RNA-seq*) obtenidos de biopsias líquidas y mostrada en el trabajo de (Aparicio-Puerta et al., 2019). ***liqDB*** ofrece una compilación de perfiles de expresión de miARN derivados de 1607 muestras, proporcionando herramientas para análisis reproducibles, comparación de resultados y generación de hipótesis. Estos perfiles se pueden explorar mediante una interfaz web que permite consultar matrices de expresión, realizar análisis de expresión diferencial, identificar miARNs más estables, y visualizar los resultados a través de gráficos. Se han utilizado dos conjuntos de datos: el primero, ***miRNA\_RCadj***, contiene los perfiles de expresión de miARN medidos en diferentes biofluidos a través de conteos de lecturas (*reads counts*). El segundo, ***metadata***, incluye información detallada sobre los experimentos, como el tipo de muestra, el sexo, el tipo de fluido, el estado de salud del paciente (cáncer o sano), entre otros atributos. Estos datos se han empleado para desarrollar modelos de clasificación capaces de detectar cáncer basándose en la expresión de miARN.

Este trabajo se estructura de la siguiente manera: en primer lugar, se presentará un marco teórico que incluirá la biología del cáncer, el papel de los miARNs, las biopsias líquidas y los fundamentos de los modelos de clasificación en ciencia de datos. Posteriormente, se discutirá el estado del arte, enfatizando los avances recientes en la investigación de miARNs como biomarcadores en el cáncer y el uso de técnicas de ciencia de datos en este ámbito, así como los desafíos técnicos encontrados en la implementación de los modelos.

A continuación, en el desarrollo del proyecto, se detallará la metodología utilizada, comenzando con el planteamiento del problema y seguido de un análisis exploratorio de los datos. Se describirán los pasos de preprocesamiento de los datos, que incluyen la normalización, el tratamiento de valores faltantes y atípicos, y el balanceo de clases. También se explicará la selección de características, que permitió identificar los miARNs relevantes. Después, detallará la optimización realizada en los cinco modelos de clasificación: SVM, KNN, XGBoost, *Random Forest* y MLP. El rendimiento de los modelos se evaluó utilizando métricas clave como AUC-ROC y F1-Score. Finalmente, se presentarán los resultados obtenidos, que destacaron a XGBoost como el modelo más eficaz.

Por último, se expondrán las conclusiones derivadas de este estudio y se sugerirán posibles líneas de trabajo futuro que podrían mejorar los resultados alcanzados y ampliar el uso de las biopsias líquidas en la detección temprana del cáncer.

## 2. Objetivos

Este apartado detalla los logros que se pretenden alcanzar a lo largo del proyecto, estructurados en un objetivo general y varios objetivos específicos que guiarán el desarrollo de la investigación.

El **objetivo principal** de este proyecto es desarrollar un modelo de clasificación basado en datos de expresión de miARN obtenidos a partir de biopsias líquidas, con el fin de detectar de manera temprana la presencia de cáncer en pacientes. Para ello, se emplearán técnicas de ciencia de datos y aprendizaje automático que permitan identificar patrones asociados con la enfermedad.

### Objetivos específicos:

1. **Análisis de la relación entre miARNs y estado de salud:** Explorar los perfiles de expresión de miARN para identificar posibles patrones o diferencias entre individuos con cáncer y sanos, determinando si existe una asociación significativa entre estos perfiles y el estado de salud.
2. **Selección de miARNs:** Realizar un análisis de las características del conjunto de datos e identificar un subconjunto de miARNs que estén asociados con la presencia de cáncer, utilizando técnicas de selección de características basados en filtros, en *wrapper* y en *embedding*.
3. **Implementación y evaluación de modelos de clasificación:** Desarrollar, entrenar y optimizar varios modelos de clasificación evaluando su rendimiento con métricas adecuadas para determinar qué modelo es el más adecuado para la detección temprana de cáncer.

## 3. Marco teórico

Este apartado desarrolla los fundamentos teóricos necesarios para contextualizar el proyecto, abordando la biología del cáncer, el rol de los miARNs, el uso de biopsias líquidas y los métodos de ciencia de datos aplicados a datos biomédicos.

### 3.1. El cáncer

El cáncer es una enfermedad que se caracteriza por el crecimiento descontrolado de las células, con capacidad para invadir tejidos cercanos y extenderse a otras partes del cuerpo, lo que se conoce como metástasis. En condiciones normales, las células tienen un ciclo de vida regulado, dividiéndose y muriendo en un equilibrio constante. Sin embargo, en el cáncer, las mutaciones genéticas o alteraciones epigenéticas provocan que este equilibrio se rompa, lo que da lugar a la proliferación descontrolada de células, que pueden formar tumores. Los tumores pueden ser benignos, sin capacidad de invasión ni metástasis, y malignos, que son los verdaderos causantes del cáncer letal (Weinberg, 2023).

A nivel molecular, los principales actores involucrados en el desarrollo del cáncer son los oncogenes, que promueven el crecimiento celular, y los genes supresores de tumores, que lo inhiben. Cuando los oncogenes se activan de manera anormal, o los genes supresores de tumores dejan de funcionar correctamente, se facilita el desarrollo del cáncer. Como resultado, las células cancerosas pierden el control de su ciclo y la capacidad de morir de manera programada apoptosis (Hanahan & Weinberg, 2011).

### 3.2. Biología molecular de la célula

La célula eucariota es la unidad básica de los organismos multicelulares y contiene en su núcleo el material genético en forma de ácido desoxirribonucleico (ADN). El ADN está organizado en cromosomas y contiene los genes, que son las instrucciones para el funcionamiento de la célula. Los genes pueden ser codificantes, cuando contienen información para producir proteínas, o no codificantes, como es el caso de los miARNs, que regulan la expresión de otros genes (Alberts et al., 2022).

La expresión génica es el proceso mediante el cual se traduce la información contenida en los genes en productos funcionales, principalmente proteínas. Este proceso está regulado con diferentes mecanismos, incluyendo la regulación post-transcripcional, donde los miARNs juegan un papel clave (Watson et al., 2013). En el contexto del cáncer, la regulación de la expresión génica se ve alterada, lo que contribuye a la proliferación celular descontrolada (Hanahan & Weinberg, 2011).

### 3.3. miARN (microARN)

Los miARNs son pequeños ARNs no codificantes que desempeñan un papel fundamental en la regulación de la expresión génica. Durante la regulación post-transcripcional, actúan uniéndose a secuencias específicas en los ARN mensajeros (ARNm), lo que provoca la degradación del ARNm o la inhibición de su traducción a proteínas. De esta manera, los miARNs regulan procesos celulares clave como el crecimiento, la diferenciación y la apoptosis (Bartel, 2004).

Los miARNs pueden estar desregulados, lo que significa que su expresión puede estar aumentada o disminuida. Esta desregulación puede favorecer el crecimiento tumoral: los miARNs que actúan como supresores de tumores pueden estar subexpresados, permitiendo que los oncogenes se activen, mientras que los miARNs que inhiben genes supresores de tumores pueden estar sobreexpresados, contribuyendo al desarrollo del cáncer (Esquela-Kerscher & Slack, 2006). Por esta razón, los miARNs son considerados importantes biomarcadores en la oncología, ya que su perfil de expresión puede proporcionar información valiosa para la detección y monitoreo del cáncer (Calin & Croce, 2006).

### 3.4. Biopsias líquidas

Las biopsias líquidas son una técnica no invasiva que permite detectar y monitorear el cáncer a través de la identificación de componentes tumorales presentes en fluidos corporales. Entre los componentes que se analizan en las biopsias líquidas se encuentran el ADN tumoral circulante (ADNct) y los miARNs, que pueden ser liberados por las células tumorales (Crowley et al., 2013; Diaz & Bardelli, 2014).

El uso de miARNs en biopsias líquidas es especialmente prometedor porque estos pequeños ARN son estables en la sangre y su perfil de expresión puede proporcionar una visión precisa del estado del cáncer en un paciente (Mitchell et al., 2008). Esto permite una detección temprana y un monitoreo continuo de la enfermedad, lo que es crucial para un tratamiento eficaz. Además, a diferencia de las biopsias de tejido, que requieren una intervención quirúrgica, las biopsias líquidas se obtienen mediante una muestra de sangre o fluido, lo que las hace menos invasivas y más seguras para los pacientes (Schwarzenbach et al., 2011; Wan et al., 2017).

### 3.5. Tecnología RNA-seq y análisis de datos de miARNs

La secuenciación de ARN (RNA-seq) es una tecnología que permite analizar el transcriptoma completo, es decir, el conjunto de ARNs presentes en una célula o tejido en un momento específico; por ello, la RNA-seq proporciona una instantánea de la expresión genética en la muestra. A través de esta técnica, no solo se cuantifican los niveles de expresión de ARN, sino que también se obtienen sus secuencias exactas, lo

que facilita la identificación de variaciones en moléculas pequeñas, como los miARNs (Conesa et al., 2016; Z. Wang et al., 2009).

Por lo tanto, Los *datasets* de miARNs generados mediante RNA-seq proporcionan una valiosa fuente de información para el desarrollo de modelos predictivos en la detección de cáncer. A través del análisis computacional de estos datos, es posible identificar patrones de expresión anómalos en miARNs que se asocian con el desarrollo tumoral, permitiendo así su clasificación como posibles biomarcadores (Li & Kowdley, 2012; Maher et al., 2009).

El proceso de RNA-seq comienza con la extracción del ARN total de la muestra biológica. A continuación, se seleccionan los tipos específicos de ARN de interés, como el ARN mensajero (ARNm) o los miARNs, dependiendo del objetivo del estudio. Posteriormente, el ARN seleccionado se convierte en ADN complementario (ADNc) mediante retrotranscripción, formando así una biblioteca de ADNc (Griffith et al., 2015). Esta biblioteca, que representa fielmente las moléculas de ARN presentes en la muestra, se secuencia utilizando plataformas avanzadas como Illumina, PacBio u Oxford Nanopore Technologies. Estas tecnologías generan un volumen masivo de datos que luego son procesados para cuantificar con precisión los niveles de expresión de los miARNs.

### 3.6. Técnicas de normalización

La cuantificación de la expresión genética en RNA-seq se realiza contando el número de lecturas (*reads counts*) que se alinean a una región específica del genoma durante el ensamblaje del transcriptoma. Este recuento directo de lecturas permite estimar la cantidad de ARN expresado en un gen o transcripto particular. Sin embargo, para poder comparar correctamente estos datos entre genes y muestras, es necesario convertir estos recuentos en métricas adecuadas para análisis posteriores, como pruebas de hipótesis o modelos de regresión (Evans et al., 2018; Robinson & Oshlack, 2010).

Existen varios factores que influyen en esta conversión:

- **Profundidad de secuenciación:** Se refiere al número total de lecturas generadas durante un experimento de secuenciación para una muestra dada. Una mayor profundidad de secuenciación corresponde a una mejor detección de transcritos (secuencia de ARN producida a partir a través de la transcripción), ya que el número de lecturas aumenta la fiabilidad y precisión de los resultados. La cantidad total de lecturas generadas varía entre experimentos. Para corregir esta variabilidad, los recuentos de lecturas se normalizan a lecturas por millón de lecturas mapeadas (RPM). Esto asegura que los resultados no estén sesgados por la cantidad total de datos generados en cada experimento.
- **Longitud del gen:** Se refiere a la extensión del ADN que constituye un gen. Los genes largos tienen una mayor cantidad de secuencias de ADN, por lo que, durante la secuenciación, se obtienen más lecturas simplemente por el hecho de que hay más secuencia disponible. Para evitar que esto genere un sesgo en la

cuantificación, se normaliza el número de lecturas dividiendo por la longitud del gen, lo que resulta en las métricas FPKM (Fragmentos por Kilobase de transcripto por Millón de lecturas mapeadas) / RPKM (Lecturas por Kilobase de transcripto por Millón de lecturas mapeadas) y TPM (Transcritos por Millón). Esto ajusta las diferencias entre genes largos y cortos, permitiendo comparaciones más precisas.

- **Cantidad total de ARN en la muestra:** De cada muestra se extrae la misma cantidad de ARN, por lo que las muestras con más ARN total tendrán menos ARN por gen. Para corregir este efecto, se aplican métodos de normalización que ajustan estas diferencias, como el método cuantil o algoritmos específicos como DESeq2, TMM (Trimmed Mean of M-values), los cuales comparan un subconjunto de genes no diferencialmente expresados y ajustan las muestras en función de esos genes de referencia.

Estos métodos de normalización son cruciales para asegurar que las diferencias observadas en los niveles de expresión de los miARNs reflejen cambios biológicos reales. Una vez normalizados, y dependiendo de la normalización elegida, los datos pueden ser utilizados para análisis estadísticos en la misma muestra (FPKM, RPKM, o TPM) o entre diferentes muestras (DESeq2, o TMM).

A continuación, se presenta cada método en detalle:

#### Lecturas por Millón (RPM)

$$FPM = \frac{\text{Número de lecturas mapeadas de un gen}}{\text{Número total de lecturas en la muestra}} \times 10^6$$

Existe una variante de la fórmula en la que se miden fragmentos en vez de lecturas (FPM).

#### Fragmentos Por Kilobase por Millón de fragmentos mapeados (FPKM)

$$FPKM = \frac{\text{Número de fragmentos mapeados a un gen}}{\text{Longitud del gen en kilobases} \times \frac{\text{Número total de fragmentos mapeados}}{10^6}}$$

Método explicado por Trapnell et al. (2010).

#### Lecturas Por Kilobase por Millón de lecturas mapeadas (RPKM)

$$RPKM = \frac{\text{Número de lecturas mapeadas a un gen}}{\text{Longitud del gen en kilobases} \times \frac{\text{Número total de lecturas mapeadas}}{10^6}}$$

Método explicado por Mortazavi et al. (2008).

#### Transcritos por Millón (TPM)

Este método es similar a los dos anteriores, pero con una diferencia clave: primero se normaliza la longitud del gen y luego se normaliza por el número total de lecturas (Wagner et al., 2012). Esto lo hace más comparable entre muestras.

$$TPM = \frac{RPLM \text{ de un gen}}{\sum(RPKM \text{ de todos los genes})} \times 10^6$$

#### *Trimmed Mean of M-values (TMM)*

TMM ajusta por diferencias en la composición de la biblioteca entre muestras, por lo que permite comparaciones más precisas entre ellas. El enfoque TMM se basa en la comparación de las distribuciones de los M-valores (log-ratios de los niveles de expresión entre dos muestras) tras eliminar los genes extremos o altamente expresados que podrían distorsionar la normalización. A través de este proceso, se eliminan valores atípicos y genes diferencialmente expresados, calculando así un factor de normalización que corrige las variaciones globales en el tamaño de la biblioteca y la composición génica entre las muestras (Robinson & Oshlack, 2010).

#### *Differential Expression Analysis with Shrinkage Estimation for Sequencing Data (DESeq2)*

DESeq2 utiliza una estimación de dispersión empírica y un modelo estadístico de distribución negativa binomial para normalizar los conteos y detectar genes diferencialmente expresados (Love et al., 2014).

### 3.7. Modelos de clasificación aplicados a datos biomédicos

Los modelos de clasificación son una técnica de aprendizaje supervisado ampliamente utilizada en ciencia de datos para asignar una etiqueta a una muestra, en función de sus características (James et al., 2023). En el ámbito biomédico, estos modelos son fundamentales para la detección de enfermedades como el cáncer.

A continuación, se presentan algunos de los algoritmos de clasificación más utilizados en conjuntos de datos biomédicos (Hastie et al., 2009; Kourou et al., 2015).

#### *Máquinas de vectores de soporte (del inglés support vector machines, SVM)*

Este algoritmo busca el hiperplano óptimo que separa las clases en el espacio de características (Cortes et al., 1995). Destaca por su capacidad de manejar datasets de alta dimensionalidad y detectar márgenes entre clases. Sin embargo, necesita técnicas de normalización y puede ser computacionalmente intensivo.

#### *K vecinos más cercanos (del inglés k-nearest neighbors, KNN)*

Este método clasifica una muestra según la mayoría de las etiquetas de sus vecinos más cercanos (Cover & Hart, 1967). Es conocido por su simplicidad, ya que no requiere una fase de entrenamiento extensa. Además, funciona bien en casos donde no se asume una distribución específica de los datos. Su sensibilidad al ruido y la dificultad para escalar con grandes volúmenes de datos son sus principales limitaciones.

#### *Árboles de Decisión (en inglés decision tree, DT)*

Este algoritmo utiliza una estructura en forma de árbol para tomar decisiones basadas en las características de los datos. Su principio fue establecido en el trabajo sobre el

algoritmo CART (Breiman et al., 1984). Son fáciles de interpretar y eficientes para descubrir interacciones complejas entre características.

#### *Bosques aleatorios (en inglés random forest, RF)*

Algoritmo de ensamble basado en árboles de decisión que mejora la precisión mediante el uso de múltiples modelos (Breiman, 2001). Es robusto frente al sobreajuste y maneja bien datos con muchas variables y con relaciones no lineales. Su principal desventaja es su coste computacional en modelos con muchos árboles.

#### *Potenciación del gradiente (en inglés gradient boosting, GB):*

Este método construye modelos de manera secuencial, donde cada nuevo modelo intenta corregir los errores del anterior (Friedman, 2002). Es eficiente en la mejora de precisión mediante la optimización de funciones de pérdida. Con el objetivo de mejorar el algoritmo, se han desarrollado implementaciones más avanzadas:

- **Extreme Gradient Boosting, XGBoost:** Algoritmo de boosting basado en árboles y diseñado para ser más eficiente y altamente escalable. Proporciona un marco de regularización de potenciación de gradiente (T. Chen & Guestrin, 2016).
- **Light Gradient Boosting Machine, LightGBM:** Versión optimizada de *gradient boosting* que se enfoca en la eficiencia y velocidad, especialmente en grandes volúmenes de datos (Ke et al., s. f.).
- **CatBoost:** Algoritmo de *boosting* desarrollado para manejar de manera eficiente variables categóricas (Prokhorenkova et al., 2017).

#### *Naive Bayes*

Basado en el teorema de Bayes y la probabilidad condicional, este clasificador asume que las características son independientes (Duda & Hart, 1973). Destaca por ser simple y eficiente, especialmente cuando se asumen características independientes.

#### *Redes neuronales artificiales (del inglés artificial neural networks, ANN) / Perceptrón Multicapa (del inglés multilayer perceptron, MLP)*

Estas estructuras están inspiradas en el funcionamiento del cerebro humano y han sido utilizadas para resolver problemas complejos de clasificación. El concepto fue desarrollado inicialmente por McCulloch & Pitts (1943) y ha evolucionado con el tiempo. Al ser capaces de aprender representaciones complejas de los datos, las redes neuronales son una opción poderosa para problemas no lineales. Sin embargo, requieren un ajuste preciso de los hiperparámetros y grandes volúmenes de datos.

#### *Regresión logística (en inglés logistic regression, LG)*

Técnica estadística para modelar la relación entre una variable binaria y un conjunto de variables predictoras. Se basa en la función sigmoide, transformando los valores predichos en probabilidades (Stevens et al., 1959).

La comprensión de estos algoritmos es fundamental para el análisis de datos de miARNs, ya que su aplicación puede facilitar la identificación de patrones que indican la

presencia de cáncer. Para una implementación práctica en Python, (Géron, 2017) ofrece una guía completa sobre cómo aplicar estos modelos.

### 3.8. Evaluación de modelos

En este apartado se describen las métricas más utilizadas para evaluar el rendimiento de los modelos de clasificación (Hastie et al., 2009). En las fórmulas se han empleado las abreviaturas Verdadero Positivo (*True Positive*, TP), Verdadero Negativo (*True Negative*, TN), Falso Positivo (*False Positive*, FP) y Falso Negativo (*False Negative*, FN).

#### *Exactitud (accuracy)*

La exactitud es el porcentaje de predicciones correctas realizadas por el modelo sobre el total de casos. En un contexto de detección de cáncer, una alta exactitud indica que el modelo identifica correctamente tanto los casos positivos (con cáncer) como los negativos (sin cáncer). Sin embargo, no es siempre la métrica más adecuada si el conjunto de datos está desbalanceado.

$$\text{Exactitud} = \frac{TP + TN}{TP + TN + FP + FN}$$

#### *Precisión (precision)*

La precisión mide la proporción de verdaderos positivos entre todas las instancias clasificadas como positivas. Es útil en contextos donde es importante minimizar los falsos positivos. En la detección de cáncer, una baja precisión implicaría que el modelo está generando demasiados falsos positivos, lo que podría generar preocupación innecesaria en los pacientes.

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

#### *Sensibilidad (recall)*

La sensibilidad mide la capacidad del modelo para identificar correctamente los casos positivos, es decir, qué porcentaje de los casos con cáncer son detectados correctamente. Es fundamental en la detección de enfermedades, ya que una baja sensibilidad significaría que muchos casos de cáncer no son detectados.

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

#### *F1-Score*

El F1-Score es la media armónica entre la precisión y la sensibilidad. Es una métrica que proporciona un equilibrio cuando se requiere un buen rendimiento tanto en términos de falsos positivos como de falsos negativos. En modelos para la detección de cáncer, es útil cuando se busca un balance entre identificar correctamente los casos positivos sin aumentar excesivamente los falsos positivos.

$$F1 = 2 \times \frac{\text{Precisión} \times \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}}$$

#### AUC-ROC (Área bajo la curva ROC)

La curva ROC (*Receiver Operating Characteristic*) representa el comportamiento del modelo a diferentes umbrales de clasificación, graficando la tasa de verdaderos positivos frente a la tasa de falsos positivos. El AUC-ROC mide el área bajo esta curva. Un valor de AUC cercano a 1 indica que el modelo es excelente para separar correctamente los casos positivos y negativos. En la detección de cáncer, es deseable un AUC-ROC alto para asegurar que el modelo puede distinguir bien entre pacientes con cáncer y sin cáncer.

#### AUC Precision-Recall

El AUC de la curva *Precision-Recall* es útil cuando el *dataset* está desbalanceado, donde los casos positivos suelen ser menos frecuentes. En estos casos, la curva ROC puede ser menos informativa. Este AUC mide el área bajo la curva de precisión frente a la sensibilidad, lo que proporciona una visión más detallada del rendimiento en los casos positivos.

### 3.9. Fuga de datos (*data leakage*)

En el ámbito de la estadística y el aprendizaje automático, la fuga de datos se refiere al uso de información que no estaría disponible en el momento de realizar predicciones futuras durante el proceso de entrenamiento del modelo. Esta situación puede llevar a una sobreestimación de la capacidad predictiva del modelo, ya que incorpora datos que, en un entorno real, no estarían accesibles. La fuga de datos suele ser sutil y puede manifestarse de diversas formas, lo que dificulta su detección y corrección. Entre las causas más comunes se encuentran la inclusión de variables que están directamente relacionadas con la variable objetivo o la contaminación de los conjuntos de entrenamiento y prueba mediante el intercambio inadecuado de información entre ellos. Este tipo de fugas puede resultar en la selección de modelos no óptimos, ya que el rendimiento aparenta ser mejor de lo que realmente es cuando se aplica el modelo en producción (Kaufman et al., 2012; Kelleher et al., 2015).

Para poder evitarlo, es importante tener en cuenta como se realiza el preprocesamiento de datos. Siempre y cuando un paso no dependa de la distribución de los datos y no esté relacionado ni utilice la variable objetivo, no introducirá *data leakage*, por lo que podrá realizarse antes de la división de datos.

Por otra parte, los algoritmos que involucren ajustar la instancia de una clase (p.ej. KNNImputer) deberán aplicarse después de la división de datos. El ajuste de los parámetros debe realizarse en el conjunto de entrenamiento, basándose exclusivamente en esos datos. Después, se puede utilizar el objeto previamente ajustado para transformar tanto el conjunto de entrenamiento como el de prueba. De esta manera, se preserva la integridad del proceso de análisis y se evita que la

información proveniente del conjunto de prueba introdujera un sesgo artificial en el modelo.

## 4. Estado de Arte

### 4.1. MiARNs como biomarcadores en el cáncer

Tal y como se ha explicado, los miARNs juegan un papel crucial en la regulación génica y están involucrados en diversas funciones celulares. En el cáncer, la desregulación de los miARNs está ampliamente asociada con la oncogénesis, promoviendo la proliferación celular y evitando la apoptosis. Estudios clásicos como los de Calin & Croce (2006) y Esquela-Kerscher & Slack (2006) demostraron que los perfiles de miARNs pueden distinguir células tumorales de las sanas.

Uno de los aspectos más destacados de los miARNs es su estabilidad en fluidos corporales como la sangre, lo que los convierte en candidatos ideales para su uso en biopsias líquidas. Los estudios realizados por Mitchell et al. (2008) y Schwarzenbach et al. (2011) mostraron que los miARNs circulantes pueden detectarse en las primeras etapas del cáncer, lo que abre la puerta a su uso como herramientas diagnósticas tempranas.

Recientemente, estudios como el de Cardinali et al. (2022) han utilizado perfiles de miARNs para predecir la progresión del cáncer, subrayando su valor no solo para el diagnóstico sino también para el pronóstico y el seguimiento de la enfermedad. Estos avances refuerzan la idea de que los miARNs son biomarcadores esenciales en la medicina.

### 4.2. Biopsias líquidas en la detección del cáncer

Las biopsias líquidas han revolucionado la forma en que se aborda el diagnóstico del cáncer. En lugar de recurrir a procedimientos invasivos, se puede obtener información crítica a partir de muestras de sangre, lo que representa un enfoque menos agresivo para los pacientes. Según Crowley et al. (2013) y Diaz & Bardelli (2014), las biopsias líquidas permiten detectar mutaciones, alteraciones epigenéticas y biomarcadores circulantes, como los miARNs y el ADNct.

Aunque las biopsias líquidas ofrecen ventajas significativas, también enfrentan desafíos. Uno de los principales retos es la sensibilidad para detectar cantidades mínimas de material tumoral circulante en fases tempranas de la enfermedad. En comparación con las biopsias tradicionales de tejido, las biopsias líquidas son menos invasivas y permiten monitorear la progresión del cáncer con mayor frecuencia, aunque pueden ser menos precisas en la identificación de la heterogeneidad intratumoral (Schwarzenbach et al., 2011).

Investigaciones recientes han aplicado biopsias líquidas en diferentes tipos de cáncer, mostrando su potencial para detectar y monitorear enfermedades como el cáncer de mama, pulmón y colon (Keup et al., 2018; Martins et al., 2021).

### 4.3. Tecnologías de secuenciación y análisis de miARNs

La tecnología RNA-seq ha sido clave en el análisis de miARNs. A lo largo de los años, las tecnologías de secuenciación han evolucionado desde métodos basados en microarrays hasta el RNA-seq, que permite un análisis más profundo y preciso de la expresión de miARN. Según Conesa et al. (2016), El RNA-seq ha revolucionado el estudio de los miARNs, facilitando la identificación de nuevos biomarcadores.

El procesamiento de datos de RNA-seq es crucial para obtener resultados fiables. Luecken & Theis (2019) y Z. Wang et al. (2009) revisan diversas técnicas de análisis, donde es necesario un preprocesamiento minucioso para eliminar sesgos y asegurar la calidad de los resultados. También se han desarrollado softwares con métodos computacionales para procesar estos datos (Amezquita et al., 2019).

Además, bases de datos públicas como *lilqDB* (Aparicio-Puerta et al., 2019) han permitido realizar análisis a gran escala, facilitando el acceso a perfiles de miARNs en distintos contextos clínicos. Estos recursos son fundamentales para la investigación en cáncer y han permitido identificar nuevos biomarcadores que pueden ser validados clínicamente.

### 4.4. Modelos de clasificación aplicados a la detección de cáncer

El uso de modelos de aprendizaje automático ha transformado el análisis de datos biomédicos. En particular, los algoritmos de clasificación han demostrado ser muy efectivos para identificar patrones en datos de expresión de miARN. Kourou et al. (2015) destacan que técnicas como SVM, árboles de decisión y redes neuronales artificial son frecuentemente utilizadas para la clasificación de cáncer a partir de perfiles de miARNs.

Por otro lado, existen estudios donde se ha realizado una detección del cáncer mediante modelos de clasificación. Ejemplo de ello son Sarkar et al. (2021), donde se propone un modelo de clasificación que integra métodos de selección de características y técnicas de ML para identificar biomarcadores de miARNs asociados a subtipos de cáncer de mama; Andreini et al. (2022), que proponen un método de clasificación en dos etapas que utiliza clasificadores específicos para diferenciar entre muestras tumorales y sanas (SVM), así como para identificar subtipos de cáncer de mama (*random forest*); y Casalino et al. (2023), que presentan un estudio de clasificación que utiliza métodos de ML como ANN, *extremely randomized trees* y *random forest* para distinguir

automáticamente entre esclerosis múltiple pediátrica y niños sanos, basado en los perfiles de expresión de miARN.

En los últimos años, las redes neuronales y el aprendizaje profundo han ganado popularidad en el análisis de datos biomédicos. Estos modelos han mostrado un alto rendimiento en la clasificación de cáncer debido a su capacidad para extraer características complejas de los datos, como destaca Alharbi & Vakanski (2023).

#### 4.5. Técnicas de procesamiento y normalización de datos RNA-seq

El procesamiento de datos RNA-seq comienza con pasos como el control de calidad y la eliminación de contaminantes, que son esenciales para garantizar la fiabilidad de los resultados (Anders et al., 2013). Estos pasos iniciales son críticos, ya que cualquier sesgo en los datos puede afectar significativamente el rendimiento de los modelos (Luecken & Theis, 2019). Es por esto por lo que realizar un análisis exploratorio de datos (EDA) es tan importante en el procesamiento de RNA-seq (Sherafatian, 2018).

En una gran cantidad de trabajos se realiza un filtrado previo de miARNs que no estén expresados (Pant et al., 2019; Sarkar et al., 2021). En la literatura también se resalta la necesidad de realizar sobremuestreo con técnicas como SMOTE para poder lidiar con el desbalanceo de clases (Andreini et al., 2022; Casalino et al., 2023; Sherafatian, 2018).

Los *datasets* de RNA-seq suelen contener valores faltantes debido a errores en la secuenciación o limitaciones en la detección de pequeñas cantidades de ARN. La imputación de valores faltantes es crucial para evitar sesgos y mejorar la calidad de los resultados. Existen varias estrategias para tratar estos valores faltantes, que van desde la imputación simple, como la imputación por la media o la mediana, hasta métodos más complejos basados en modelos (Stekhoven & Bühlmann, 2012; van Buuren & Groothuis-Oudshoorn, 2011).

La normalización es otra etapa crucial en el análisis de datos de miARNs. Métodos como RPKM, TPM, TMM, o DESeq2 permiten ajustar las diferencias en la profundidad de secuenciación entre muestras, lo que asegura que los datos sean comparables entre sí. (Bullard et al., 2010; Tam et al., 2015) demuestran que una normalización adecuada puede mejorar el rendimiento de los modelos de clasificación y ayudar a detectar diferencias reales en la expresión de miARN.

En la literatura se han realizado diferentes estudios que evalúan y comparan estos métodos. Aunque existen ciertos estudios que establecen que la elección del método depende de las características del *dataset* (Conesa et al., 2016; Weiss et al., 2017), otros estudios señalan que métodos como DESeq2 y TMM son más apropiados para analizar RNA-seq (Dillies et al., 2013; Zhao et al., 2021). Para determinar el mejor método de normalización, incluso se ha diseñado un algoritmo por C. Wang et al. (2020).

## 4.6. Selección de biomarcadores de miARN

La selección de características es fundamental en los estudios de RNA-seq, especialmente en datos de alta dimensionalidad donde el número de características (miARNs) es superior al número de observaciones (muestras), como los perfiles de miARNs en biopsias líquidas. Identificar los miARNs más relevantes para la clasificación de cáncer reduce la complejidad del modelo y mejora su precisión (Alharbi & Vakanski, 2023).

Entre las técnicas clásicas destacan las pruebas estadísticas de filtrado como la Información Mutua y ANOVA, que permiten identificar miARNs expresados entre grupos de pacientes, como sanos versus cáncer. Pant et al. (2019) emplean técnicas de filtrado como *conditional mutual information maximisation* (CMIM), *double input symmetrical relevance* (DISR), *interaction capping* (ICAP), *conditional informative feature extraction* (CIFE) para seleccionar miARNs relevantes en el cáncer de estómago. Asimismo, Sarkar et al. (2021) validan ocho métodos de selección de características de filtrado entre los que se encuentran CMIM, DISR, ICAP y CIFE para detectar miARNs implicados en el cáncer de mama.

En los últimos años, los algoritmos de selección de características de envoltura (*wrapped*) o integrados (*embedded*), como *recursive feature elimination* (REF), *random forest* y SVM, han ganado popularidad por su capacidad para manejar la alta dimensionalidad y clasificar de manera efectiva (Rezaee et al., 2022). Además, técnicas de regularización como LASSO se utilizan para eliminar miARNs irrelevantes y mejorar la interpretabilidad de los modelos (Casalino et al., 2023).

## 4.7. Estudios comparativos

La comparación de algoritmos de clasificación ha sido un área activa de investigación. Alharbi & Vakanski (2023) revisan diversos enfoques aplicados a la clasificación de miARNs en cáncer, destacando la superioridad de algunos modelos en función de su precisión y la capacidad de generalización. También existen estudios más generalistas que revisan los algoritmos más habituales en ciencia de datos y los clasifican según su caso de uso más adecuado (Fang et al., 2016). Estos estudios permiten identificar cuáles son los algoritmos más adecuados según el tipo de cáncer y los datos disponibles.

También se han realizado estudios comparativos de métodos de imputación de valores faltantes (Lin & Tsai, 2020), incluso en *datasets* médicos (Bell et al., 2014).

## 5. Metodología

En esta sección se detallan los procedimientos metodológicos seguidos para llevar a cabo el análisis de los datos de expresión de miARN en biopsias líquidas y la construcción de modelos para la detección de cáncer. El enfoque se ha estructurado en varias fases, desde la carga y preprocesamiento de los datos, hasta la construcción y evaluación de los modelos de clasificación. Cada paso fue implementado utilizando herramientas de análisis de datos y aprendizaje automático, integradas en un entorno de desarrollo Python y R.

El código fuente completo de este proyecto, incluyendo todos los scripts utilizados para los análisis, está disponible en un repositorio público de GitHub, proporcionando transparencia y replicabilidad a los resultados obtenidos. Este es el enlace que da acceso: <https://github.com/mnunezmartinez/14MBID---TFM>.

### 5.1. Carga de datos

En este proyecto se utilizan dos conjuntos de datos principales extraídos de la base de datos *liqDB* (Aparicio-Puerta et al., 2019): *miRNA\_RCadj*, que contiene los perfiles de expresión de miARN en diferentes biofluidos, y *metadata*, que incluye la información adicional de los experimentos. Ambos *datasets* se encuentran en formato TXT, y su lectura se llevó a cabo utilizando la librería Pandas, ampliamente utilizada en la manipulación de datos en Python. A partir de este momento, los *datasets* serán nombrados como *mirna* y *metadata*.

La base de datos *liqDB* se generó a partir de datos recolectados del repositorio *NCBI Short Read Archive* (SRA) y publicaciones en *PubMed*. Los archivos originales se descargaron en formato SRA y se convirtieron a formato FASTQ utilizando fastq-dump. Para el perfilado de expresión, se empleó sRNAbench, que asigna recuentos de lectura a cada secuencia única tras su limpieza y mapeo al genoma humano GRCh38 y bases de datos de secuencias bacterianas y virales mediante bowtie1. Los niveles de expresión de los miARNs se calcularon en función de los recuentos de lectura ajustados por mapeo múltiple.

El *dataset* *mirna* contiene los niveles de expresión de 2419 miARNs (filas) en 1559 muestras (columnas). Este *dataset* se cargó en un dataframe de pandas, donde la columna *name* representa los nombres de los miARNs, y cada columna subsiguiente corresponde a un experimento específico identificado por un código único (SRX).

En cuanto al *dataset* *metadata*, se cargó de la misma manera, permitiendo asociar cada experimento con sus características experimentales. En este caso, el *dataset* contiene 1606 muestras, un número mayor que en *mirna*. Con el objetivo de entender qué valores podía contener cada característica, se realizó una revisión de los valores únicos, los cuales se detallan en la siguiente lista:

- **SRP:** Identificador del estudio.
- **Experiment:** Identificador del experimento asociado.
- **Sample:** Identificador de la muestra.
- **Instrument:** Instrumento utilizado para la secuenciación: Illumina HiSeq 2000, NextSeq 500, Illumina Genome Analyzer, Illumina HiSeq 4000, Illumina HiScanSQ, Illumina HiSeq 2500, Illumina Genome Analyzer IIx, Illumina Genome Analyzer II, Illumina MiSeq.
- **Sex:** Sexo del individuo de la muestra: female, male
- **Fluid:** Tipo de biofluido: plasma, serum, blood cells, whole blood, urine, cerebrospinal fluid, urine cells, exosomes, breast milk, seminal fluid, bile, amniotic fluid, bronchoalveolar lavage, ovarian follicle fluid, saliva, vaginal secretion, menstrual secretion, perspiration, leukocytes, full urine, full saliva
- **Extraction:** Método de extracción utilizado: miRNeasy, mirVana, miRCURY, PAXgene, TRIzol, Norgen Biotek, QiaAmp
- **Library:** Tipo de biblioteca utilizada para la secuenciación: Illumina, NEBNext, NEXTflex
- **Healthy:** Indica si la muestra proviene de un individuo sano: True, False
- **Cancer:** Indica si la muestra proviene de un individuo con cáncer: True, False
- **Exosome:** Indica si la muestra contiene exosomas: True, False
- **Desc:** Descripción adicional de la muestra.

#### Variable objetivo

El objetivo de este estudio es la clasificación de las muestras entre pacientes sanos y pacientes con cáncer. Esta información se encuentra en la columna **Cancer** del *dataset metadata*. Esta variable será utilizada en la etapa de modelado para entrenar y evaluar los modelos de clasificación, con el fin de predecir el estado de salud de los individuos basándose en los perfiles de expresión de los miARNs. Al igual que otras características del *dataset metadata*, también se utilizará para realizar ciertos procedimientos en el preprocessamiento de las expresiones de miARNs.

## 5.2. Análisis exploratorio de datos inicial

En esta sección se presentan los análisis exploratorios realizados tanto para los perfiles de expresión de miARN como para el conjunto de metadatos. El EDA es una etapa clave para entender las características de los datos, identificar posibles problemas como valores faltantes o valores atípicos, y descubrir patrones ocultos que puedan ser relevantes para el modelado posterior. A continuación, se describen los pasos seguidos.

### 5.2.1. Exploración de miARNs

#### *Estadísticas descriptivas*

En primer lugar, se calcularon las estadísticas descriptivas básicas de los experimentos, tales como la media, mediana, rango y desviación estándar. Estas métricas permitieron obtener una visión preliminar de la variabilidad de los niveles de expresión de los

miARNs entre los distintos experimentos. Las métricas se detallan en el archivo stats\_mirna\_raw.csv, que se encuentra dentro del repositorio de GitHub. La ruta completa es: results/statistics/stats\_mirna\_raw.csv

Dado que la integridad de los datos es crucial para el análisis posterior, se verificó la existencia de valores faltantes en las mediciones de expresión de miARN. Se comprobó que todos los experimentos contenían la totalidad de los 2419 registros, garantizando la ausencia de valores faltantes en el *dataset*.

Se calculó la proporción de ceros en cada experimento. Asimismo, se analizó la desviación estándar para evaluar la dispersión de los datos y detectar posibles valores atípicos o miARNs con una expresión inusualmente alta.

Para complementar el análisis, se generaron representaciones gráficas de los datos, tales como diagramas de caja y gráficos de dispersión, que permitieron visualizar tanto la distribución general como la presencia de posibles valores atípicos. A continuación, se especifica en detalle cada representación realizada.

#### *Distribución de datos*

Para evaluar la distribución de los niveles de expresión de los miARNs, se generaron histogramas y gráficos de densidad de núcleo (KDE) para 30 muestras seleccionadas aleatoriamente del *dataset*, aunque en esta memoria solo se muestren 6. Los histogramas muestran en el eje x los valores de expresión de los miARNs y en el eje y la frecuencia de aparición de dichos valores.

Se aplicó la transformación logarítmica ( $\log_2(x + 1)$ ). Adicionalmente, se generaron gráficos de KDE para verificar si se habían suavizado las distribuciones tras la transformación.

Con el fin de comprobar si los datos de expresión de los miARNs seguían una distribución normal, se aplicó la prueba de normalidad de Shapiro-Wilk a los datos transformados, obteniendo un resultado de  $p - \text{valor} = 9,63 \times 10^{-251}$ , lo que confirmaba la no-normalidad. Dado que muchos algoritmos de *machine learning* asumen normalidad en los datos, esta prueba permitió identificar la necesidad de aplicar normalizaciones adicionales o de utilizar modelos que no dependan de esta suposición.

#### *Detección de valores atípicos*

Dada la naturaleza de los datos y su alta dimensionalidad (2419 miARNs y 1559 muestras), se empleó el método estadístico del rango intercuartílico (IQR) para identificar posibles valores atípicos. Este método calcula la diferencia entre el tercer y el primer cuartil (Q3 - Q1) y considera como valores atípicos aquellos valores que se encuentran por debajo de  $Q1 - 1.5 * \text{IQR}$  o por encima de  $Q3 + 1.5 * \text{IQR}$ .

Para visualizar y corroborar la presencia de valores atípicos identificados mediante el método IQR, se generaron diagramas de caja para 30 muestras seleccionadas aleatoriamente, las mismas que en los histogramas, aunque en esta memoria solo se

muestren 6. Los diagramas de caja muestran de manera gráfica los valores atípicos que superan el rango intercuartílico.

#### *Análisis de correlación*

Para llevar a cabo el análisis de correlación entre los distintos miARNs del *dataset*, primero fue necesario transponer los datos, de manera que los miARNs se dispusieran en columnas, permitiendo las correlaciones entre cada par de miARNs. El análisis se centró en identificar pares de miARNs cuya correlación absoluta fuera mayor a 0.7 y, en un análisis más exhaustivo, mayor a 0.85.

El análisis se visualizó mediante un mapa de calor que muestra las 25 correlaciones más elevadas en valor absoluto, lo que permitió detectar los miARNs que están más estrechamente relacionados entre sí.

El siguiente paso consistió en calcular la correlación entre cada miARN y la variable objetivo Cáncer, lo que ayudó a identificar si existía una correlación real. Las 40 correlaciones más altas en términos absolutos se visualizaron gráficamente, con el fin de resaltar aquellos miARNs cuya correlación con el estado de cáncer podría ser relevante para el modelo de predicción.

#### *Análisis multivariado*

Para reducir la dimensionalidad del *dataset* de expresión de miARN y facilitar la visualización de las principales fuentes de variabilidad entre las muestras, se llevó a cabo un análisis de componentes principales. El PCA transforma las variables originales en un conjunto de componentes ortogonales (componentes principales) que capturan la mayor variabilidad posible en los datos. Este enfoque no solo permite identificar patrones entre las muestras, sino también detectar muestras que podrían no seguir el mismo comportamiento que el resto.

Para detectar valores atípicos y patrones entre los experimentos, el *dataset* de miARNs fue transpuesto. Los dos primeros componentes (PC1 y PC2) se seleccionaron para la visualización gráfica de la distribución de las muestras. Este mismo PCA también se utilizó para realizar gráficos de dispersión con los metadatos *Healthy*, *Cancer* y *Fluid*.

También se utilizó el PCA para identificar patrones de correlación entre los miARNs. En este caso, no se tuvo que transponer el *dataset*.

Finalmente, en el primer PCA se implementó el algoritmo de agrupamiento *k-means clustering*, para investigar posibles agrupamientos naturales entre las muestras en función de sus niveles de expresión de miARN. Se probaron distintos valores de k para optimizar el número de clústeres. También se utilizó el método del codo (*elbow method*) para determinar el valor más apropiado de k. Por último, se exploró la agrupación con k=2, para diferenciar entre *Healthy / No Healthy* y *Cancer / No Cancer* y K=5 para la característica *Fluid*.

## 5.2.2. Exploración de la metadata

### *Estadísticas descriptivas*

Para comenzar con la exploración de la *metadata*, se generó un resumen estadístico de las variables categóricas del *dataset* (Apéndice I: Estadísticas descriptivas). Este resumen permitió evaluar la cantidad de valores únicos y la frecuencia del valor más común en cada categoría. Además, se identificaron valores faltantes en varias de estas variables.

### *Distribución de variables*

Se utilizaron gráficos de barras para visualizar la cantidad de observaciones por cada categoría.

## 5.3. Preprocesamiento

En esta sección se describen las técnicas de preprocesamiento aplicadas tanto al conjunto de datos de *mirna* como a *metadata*. El preprocesamiento es una fase esencial para asegurar que los datos estén en una forma adecuada para el modelado, ya que permite corregir problemas como valores duplicados, baja expresión de miARN, datos faltantes y valores atípicos. Además, se abordan los métodos de normalización, estandarización y balanceo de clases, que mejoran la calidad de los datos y evitan la fuga de datos. A continuación, se detallan los pasos realizados para limpiar, transformar e integrar los datos.

### 5.3.1. Valores duplicados

Primero, se realizó un análisis de valores duplicados para poder eliminarlos y que no hubiera copias de la misma observación en el *dataset*. En este caso, no se encontraron valores duplicados en ninguno de los dos conjuntos de datos.

### 5.3.2. Filtrado de miARNs con baja expresión

En esta etapa, se realizó un filtrado de los miARNs con baja expresión para reducir el ruido en los datos y centrarse en aquellos con relevancia biológica potencial. Este paso es crucial, ya que la mayoría de los miARNs presentan recuentos extremadamente bajos o incluso nulos en muchas muestras, lo que puede dificultar el análisis posterior. Al reducir el número de características irrelevantes, se optimiza el rendimiento de los modelos de clasificación que se emplearán posteriormente.

Siguiendo el criterio de filtrado utilizado por Pant et al. (2019), se eliminaron aquellos miARNs que tenían un nivel de expresión igual a cero en al menos el 60% de las muestras.

### 5.3.3. Normalización de miARNs

Las tecnologías de secuenciación introducen variabilidad técnica (Conesa et al., 2016). Por lo tanto, los datos crudos del transcriptoma se deben normalizar para corregir las variaciones técnicas que pueden afectar la cuantificación de la expresión génica y llevar a conclusiones incorrectas (Abrams et al., 2019).

En este caso, se optó por la normalización con TMM, y se realizó utilizando el paquete *edgeR* en R (Robinson et al., 2010). Para poder realizar la normalización TMM, se eliminaron tres columnas (SRX) en el *dataset* que tenían todos los niveles de expresión a 0. Posteriormente, el *dataset* normalizado se importó nuevamente a Python para continuar con el análisis.

Tras la normalización con TMM, se volvió a evaluar la presencia de valores atípicos para determinar si la variabilidad en la expresión de los miARNs se había reducido.

Después del filtrado y la normalización, se obtuvo un *dataset* con 1557 filas (experimentos) y 415 columnas (414 miARNs).

### 5.3.4. Estandarización de datos

Tras aplicar la normalización TMM para ajustar las diferencias en la composición de la muestra y la profundidad de secuenciación, se realizó una transformación logarítmica a los datos. Esta transformación es una práctica común en estudios genéticos, especialmente cuando los datos presentan una alta dispersión y sesgo. La transformación logarítmica tiene como objetivo reducir la varianza entre los valores de expresión, estabilizar las distribuciones y hacer que los datos sean más adecuados para su análisis posterior mediante modelos estadísticos y *de machine learning* (Ewens & Grant, 2005).

### 5.3.5. Integración de datos y división en conjuntos de entrenamiento y prueba

#### *Integración de datos*

En este trabajo, se adoptó una estrategia para prevenir la fuga de datos durante el preprocessamiento y la construcción de los modelos de clasificación. Primero, y debido a que la clase objetivo *Cancer* estaba desbalanceada, se integraron los *datasets* *mirna* y *metadata* mediante la característica *Experiment*, lo que permitió realizar una estratificación adecuada. Esta integración facilitó mantener la representatividad de las clases en ambos conjuntos, garantizando que el modelo se entrena y evalúe de manera equitativa y sin sesgos inducidos por la distribución inicial de los datos.

Tras la integración, se obtuvo un *dataset* (*mirna\_metadata*) con 1556 filas (experimentos) y 427 columnas (diferentes miARNs y la metadata).

### División de datos

A continuación, se eliminaron los valores nulos de la variable *Cancer*. El número de observaciones eliminadas fue 28, un 1,80%, por lo que se consideró un porcentaje apto (inferior al 5%) que permitía eliminar las observaciones sin dañar el *dataset*.

Después, se realizó la división del *dataset* de perfiles de miARN en conjuntos de entrenamiento y prueba. Tras la división, se obtuvo un *dataset* de entrenamiento con 1222 observaciones y un *dataset* de prueba con 306 observaciones. También se obtuvieron los conjuntos de datos de *mirna* y *metadata* por separado, respetando la división realizada.

### 5.3.6. Valores faltantes en *metadata*

El proceso de limpieza de valores faltantes en *metadata* comenzó con la identificación de las columnas afectadas, tanto en el conjunto de entrenamiento como en el de prueba. Además, se utilizaron mapas de calor para verificar la distribución de los valores faltantes entre filas y columnas.

Para abordar este problema, se emplearon diferentes enfoques en función del porcentaje de valores faltantes y de la relevancia de la columna:

- **Sex:** Dado que esta columna es imprescindible para el análisis, los valores faltantes se imputaron utilizando el método de *k-nearest neighbors* (KNN), el cual infiere los valores faltantes basándose en las observaciones más similares dentro del *dataset*.
- **Fluid y Healthy:** Debido al bajo porcentaje de valores faltantes, se eliminaron las observaciones con valores nulos, ya que la eliminación de un pequeño número de filas no afecta la representatividad del conjunto de datos.
- **Extraction y Library:** Estas columnas presentaban un porcentaje moderado de valores faltantes. En el caso de querer imputar valores faltantes, se debe realizar con KNN. Sin embargo, en este trabajo, no se consideró necesario al no estar directamente relacionados con las características biológicas.
- **Desc:** Al presentar un alto porcentaje de datos faltantes y no considerarse esencial para el análisis, la columna fue eliminada del *dataset*. Además, la información de esta característica era muy diversa, por lo que no fue de utilidad.

Antes de la imputación con KNN, se llevó a cabo un análisis de valores atípicos, ya que estos pueden distorsionar las distancias en las que se basa el KNN, resultando en imputaciones incorrectas. Tanto este paso como la imputación de valores se explican en los siguientes apartados.

Tras la eliminación, se obtuvo un *dataset* de entrenamiento con 1220 observaciones y un *dataset* de prueba con 305 observaciones.

### 5.3.7. Codificación de variables categóricas en metadata

Para utilizar algoritmos que dependen de distancias, como DBSCAN y KNN, todas las variables categóricas presentes en el *dataset* deben ser transformadas en un formato numérico adecuado. Esto es debido a que estos algoritmos dependen de las relaciones espaciales entre las observaciones, y las variables categóricas no tienen un orden natural que represente estas relaciones. Por tanto, se optó por diferentes métodos de codificación según el tipo de variable.

- **One-Hot Encoding** (Géron, 2017): Se aplicó a las variables categóricas nominales (sin orden natural) como *Instrument*, *Fluid*, *Extraction*, y *Library*. *One-Hot Encoding* genera una columna binaria para cada categoría, lo que permite que los modelos de ML las interpreten correctamente. Para evitar la trampa de las variables ficticias (*dummy variable trap*), se utilizó la configuración *drop=first* en la instancia de la clase OneHotEncoder, eliminando la primera categoría de cada variable codificada, lo que evita la multicolinealidad.
- **Label Encoding**: Se utilizó para variables binarias como *Sex*, *Healthy*, *Cancer*, y *Exosome*, transformándolas en 0 (False) y 1 (True).
- **SRP, Experiment, Sample**: Estas columnas contenían identificadores únicos, que no aportan información predictiva ni son útiles para los modelos de *clustering* o imputación. Por lo tanto, se excluyeron del proceso de codificación.

### 5.3.8. Valores atípicos en miARN

Primero, y antes de aplicar los métodos de detección de valores atípicos (en inglés, *outliers*), se realizó una visualización PCA del conjunto de datos para observar si las muestras con cáncer seguían algún patrón, ya que cabía la posibilidad que el preprocesamiento anterior hubiera modificado los patrones analizados en la EDA.

Para identificar observaciones que podrían afectar el rendimiento de los modelos de clasificación y eliminar aquellas que representaran comportamientos atípicos no deseados, se implementaron y compararon tres métodos diferentes de detección de valores atípicos (Das et al., 2022):

- **Isolation Forest (IF)**: Algoritmo basado en árboles diseñado específicamente para la detección de anomalías. Tiene como objetivo identificar puntos que están aislados o son atípicos en comparación con la mayoría de los datos (Liu et al., 2008).
- **Local Outlier Factor (LOF)**: Algoritmo que mide la densidad local de un punto en relación con sus vecinos. Un punto se considera anomalía si su densidad es mucho menor que la densidad de sus vecinos, lo que indica que está "aislado" en comparación con su entorno (Breunig et al., 2000).

- **Distancias Multivariadas (Mahalanobis):** Este método calcula la distancia multivariada de cada observación en función de las características del *dataset* (Mahalanobis, 1936).

Adicionalmente, se probó con el algoritmo **DBSCAN** (Ester et al., 1996), aunque no fue utilizado en la selección final de valores atípicos.

En cada método, se probaron diferentes configuraciones de los hiperparámetros. Éstas están explicadas en detalle en el Apéndice II: Detección de valores atípicos.

Se utilizó una combinación de estos métodos para asegurar que los valores atípicos fueran consistentes entre los diferentes enfoques y así mejorar la precisión de la selección. Se seleccionaron aquellas observaciones que fueron clasificadas como *outliers* en al menos dos de los tres métodos escogidos.

Es importante destacar que la eliminación de *outliers* solo se aplicó al conjunto de entrenamiento. No se eliminaron *outliers* del conjunto de prueba, ya que este debe reflejar datos reales no vistos, incluyendo posibles observaciones atípicas. Mantener estos datos en el conjunto de prueba permite evaluar la capacidad de generalización del modelo ante nuevos datos. Si el modelo ha sido entrenado correctamente, debería ser capaz de manejar los valores atípicos presentes en el conjunto de prueba.

### 5.3.9. Valores atípicos en miARN y metadata

Tras la detección inicial de valores atípicos utilizando solo el *dataset mirna*, se decidió realizar una integración con *metadata* para un análisis más preciso de *outliers*. La combinación de ambos *datasets* permitió considerar factores adicionales que pueden influir en la expresión de los miARNs, como el estado de salud de los pacientes o los instrumentos utilizados en la secuenciación.

#### *Selección de columnas*

Para realizar el análisis de valores atípicos en los datos combinados, se seleccionaron varias columnas de *metadata* que pudieran tener un impacto en la expresión de miARNs y ayudar a detectar *outliers* con mayor precisión. Estas columnas fueron:

- **Sex:** El sexo puede afectar los perfiles de expresión de los miARNs, por lo que se consideró relevante para identificar *outliers*.
- **Healthy:** Dado que la condición de salud es un factor clave, se utilizó para distinguir perfiles de expresión relacionados con el cáncer de aquellos asociados a la salud.
- **Exosome:** Esta variable podría ser relevante para detectar si los perfiles de miARN en exosomas seguían patrones específicos o se desviaban de las expectativas.
- **Instrument:** Las diferencias entre los tipos de instrumentos utilizados podrían influir en los perfiles de expresión.

- **Fluid:** Los diferentes tipos de fluidos biológicos, como plasma, suero u orina, generaban patrones únicos en los perfiles de miARN, como se comprobó en la EDA, por lo que podrían ser clave para identificar *outliers*.
- **Extraction:** Diferentes técnicas de extracción de miARN podrían generar perfiles inusuales que también fueron considerados en el análisis.
- **Library:** Las diferentes bibliotecas podrían generar perfiles diferentes que se tuvieron en cuenta.

#### *Detección de valores atípicos*

Para detectar *outliers* en el conjunto de datos integrado, se utilizaron las técnicas de **Isolation Forest** y **Local Outlier Factor**. Ambas técnicas se aplicaron con la contaminación automática y la parametrización seleccionada en el apartado anterior.

Con el objetivo de verificar los resultados, para ambas configuraciones se realizó un gráfico con PCA. Éstos están adjuntos en el apartado Apéndice II: Detección de valores atípicos.

Los resultados de ambos algoritmos fueron cruzados, seleccionando solo aquellos valores que aparecían como valores atípicos en ambos métodos. Estos valores fueron eliminados del *dataset* final.

### 5.3.10. Imputación de valores faltantes en metadata

Se seleccionó la técnica KNN (Cover & Hart, 1967) para llevar a cabo la imputación de valores faltantes, ya que permite utilizar la relación entre diferentes muestras y variables para estimar los valores faltantes de manera precisa (Fix & Hodges, 1989). Esta técnica aprovecha las similitudes en el espacio de múltiples variables, como las expresiones de miARN y las columnas seleccionadas de *metadata*, para imputar los datos faltantes.

Las columnas que necesitaban imputar valores faltantes eran *Sex*, *Extraction* y *Library*. Se consideró que *Sex* tenía mayor relevancia biológica por su influencia en la expresión de miARNs, dado que existen diferencias biológicas entre hombres y mujeres que pueden impactar la expresión de ciertos miARNs y, por ende, la detección de enfermedades. Por otro lado, las columnas *Extraction* y *Library*, que representan procesos técnicos de extracción de ARN y bibliotecas de secuenciación, no se incluyeron en el proceso de imputación. Aunque estos factores pueden introducir variabilidad en los datos, no están directamente relacionados con las características biológicas que son clave para predecir los miARNs asociados a enfermedades. Estas dos fueron eliminadas del *dataset*.

Si los niveles de expresión de ciertos miARNs están correlacionados con características específicas en *metadata*, estos datos pueden proporcionar un contexto adicional que ayuda a imputar valores faltantes de manera más precisa. Por lo tanto, se seleccionaron las variables de *metadata* *Exosome*, *Fluid* y *Healthy* como ayuda en la imputación.

### 5.3.11. Balanceo de la clase objetivo *Cancer*

El *dataset* original presentaba un desequilibrio significativo *Cancer*, con 803 muestras negativas y 391 muestras positivas.

Entre las estrategias de sobremuestreo (*over-sampling*), submuestreo (*under-sampling*) y el ajuste de pesos en la función de pérdida, el sobremuestreo fue seleccionado como la mejor alternativa, ya que permite aumentar la cantidad de muestras de la clase minoritaria sin perder información de la clase mayoritaria.

En concreto, se utilizó SMOTE (*Synthetic Minority Over-sampling Technique*) como técnica principal para abordar el desbalanceo de la clase objetivo *Cancer* (Chawla et al., 2002). SMOTE crea nuevas muestras sintéticas interpolando los puntos de datos de la clase minoritaria, lo que introduce variaciones en los datos y mejora la capacidad del modelo para generalizar mejor los patrones de la clase minoritaria.

Después de aplicar SMOTE, se logró balancear el conjunto de entrenamiento, aumentando la cantidad de observaciones de la clase minoritaria hasta obtener 1606 observaciones en total en el conjunto de entrenamiento.

## 5.4. Análisis exploratorio de datos posprocesamiento

Con el objetivo de verificar la calidad de los datos tras el preprocesamiento, se creó un *pipeline* para realizar lo siguiente:

- Revisión de los tipos de datos: Para asegurarse de que correspondieran a su naturaleza, ya sea categórica o numérica, y que no hubiera inconsistencias en la asignación de estos.
- Resumen estadístico: Se generó un resumen estadístico de todas las variables numéricas, obteniendo medidas como la media, la mediana, la desviación estándar, los valores mínimo y máximo, y los cuartiles, con el fin de verificar la distribución y dispersión de los datos.
- Inspección de valores faltantes: Para identificar la presencia de valores faltantes, los cuales podrían haber sido omitidos durante el preprocesamiento.
- Distribución de datos mediante histogramas y KDE: Para visualizar la distribución de las variables. Esto permitió identificar patrones de asimetría, normalidad o sesgos significativos en la distribución de los datos.
- Prueba de Shapiro-Wilk: Para evaluar la normalidad de las distribuciones de los datos, se aplicó la prueba de Shapiro-Wilk a las variables numéricas.
- Detección de valores atípicos mediante análisis de rango intercuartílico (IQL) y diagramas de caja: Para detectar la presencia de valores atípicos, que podrían influir negativamente en el rendimiento del modelo si no son tratados adecuadamente.
- Correlación de características:

- Entre perfiles de miARN: Para identificar las relaciones más fuertes y débiles entre las variables. A partir de esta matriz, se generó un mapa de calor para visualizar de manera clara las correlaciones significativas.
- Entre miARN y *Cancer*: Identificar miARNs potencialmente relevantes para la clasificación.
- Gráfico de dispersión mediante PCA: Para reducir la dimensionalidad de los datos y facilitar la visualización de las relaciones entre las muestras. En el gráfico resultante, se resaltaron las muestras de cáncer para evaluar si estas se agrupan de manera coherente en el espacio de menor dimensionalidad, formando patrones.

## 5.5. Selección de miARNs

Tras el preprocesamiento, el *dataset* estaba compuesto por 416 características frente a 1606 observaciones, de las cuales 415 son miARNs; por lo que era de alta dimensionalidad. En biomedicina, la interpretabilidad puede ser muy importante; los investigadores y profesionales médicos a menudo necesitan entender qué características (miARNs) están influyendo en las predicciones (Leung et al., 2022; Srinivasu et al., 2022). Sabiendo esto, y tras haber comprobado en el EDA que el *dataset* tiene muchas características potencialmente irrelevantes o redundantes, se realizó la selección de características. Esto permitiría identificar los miARNs más importantes que contribuyen a la clasificación antes de considerar reducir dimensionalidad, si fuera necesario.

La selección de características es necesaria por varias razones. Primero, mejora la interpretabilidad al permitirnos identificar los miARNs más relevantes para la clasificación de las muestras. Segundo, ayuda a reducir el riesgo de sobreajuste al eliminar variables irrelevantes o ruidosas, mejorando la eficiencia del modelo (Hastie et al., 2009). A continuación, se describen los métodos utilizados para seleccionar las características:

### *Filtrado por varianza baja*

Si una característica tiene una varianza muy baja no aporta información valiosa, por lo que se considera irrelevante para la clasificación y puede eliminarse. Es por esto por lo que se identificaron y eliminaron las características cuya varianza era inferior a 0,01.

### *Análisis de varianza (ANalysis Of VAriance, ANOVA)*

Este método evalúa la capacidad discriminatoria de cada miARN frente a la variable objetivo *Cancer*. Se retuvieron aquellas características cuyo valor *p* era menor a 0,05.

### *Información mutua (Mutual Information, MI)*

La MI mide la dependencia entre cada miARN y la variable objetivo *Cancer*, capturando tanto relaciones lineales como no lineales. Se aplicó una curva de validación para determinar el número de características óptimas (Figura 3).

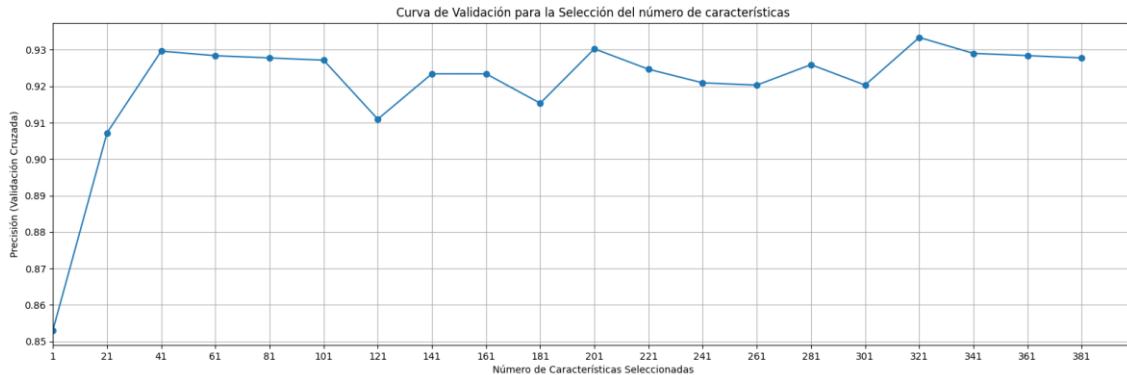


Figura 3: MI - Curva de validación para seleccionar el número de características.

#### Eliminación recursiva de características cross-validation con support vector machines (RFECV + SVM)

En este enfoque, SVM se emplea como el clasificador base, donde se ajusta el modelo utilizando un conjunto de características. RFECV, por su parte, elimina recursivamente las características menos relevantes, evaluando el rendimiento del modelo en cada paso mediante validación cruzada.

#### Support vector machines (SVM) con regularización L1

SVC es un clasificador basado en SVM, que intenta encontrar un hiperplano que separe las clases en un espacio de características de alta dimensión. Utilizar el penalizador L1 en SVC implica que el modelo aplicará una regularización L1 (*Lasso*), que fuerza algunos coeficientes a 0. Se utilizó una curva de validación (Figura 4) para ajustar el hiperparámetro C a 0,1.

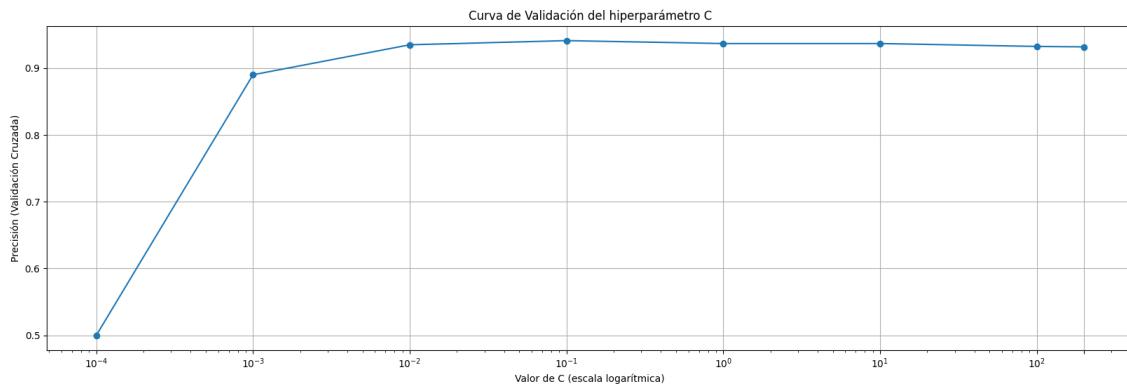


Figura 4: SVM + L1 - Curva de validación para seleccionar el valor del hiperparámetro C.

#### Random forest (RF)

Este algoritmo basado en árboles mide la importancia de cada característica durante la construcción de los árboles de decisión. Las características con mayor importancia contribuyen más a la predicción. Se aplicó una curva de validación para seleccionar el número adecuado de miARNs (Figura 5).



Figura 5: RF - Curva de validación para seleccionar el número de características.

Finalmente, las características seleccionadas en al menos tres de estos métodos se consideraron las más relevantes.

Para evaluar la selección realizada, se utilizaron los siguientes métodos:

- Comparación de las métricas de rendimiento AUC-ROC, F1-Score, precisión (*precision*) y sensibilidad (*recall*) utilizando un modelo SVN por defecto con las características seleccionadas y sin seleccionar.
- Análisis de la curva de aprendizaje.
- Análisis de correlación entre los miARNs seleccionadas.
- Comparación del rendimiento antes y después de la selección de características en términos de exactitud (*accuracy*).

## 5.6. Construcción de modelos de clasificación

Para la detección del cáncer a partir de miARNs en biopsias líquidas, se seleccionaron varios algoritmos de clasificación debido a sus capacidades para manejar datos de alta dimensionalidad y problemas de clases desbalanceadas: *support vector machines* (SVM), *k-nearest neighbors* (KNN), *random forest classifier* (RFC), XGBoost (XGB) y *multi-layer perceptron* (MLP).

Se utilizaron tanto GridSearchCV como RandomizedSearchCV para la optimización de hiperparámetros, dependiendo del caso. La primera técnica realiza una búsqueda exhaustiva de todas las combinaciones posibles, lo que garantiza la evaluación del espacio de búsqueda completo, mientras que la segunda toma muestras aleatorias dentro del espacio de hiperparámetros, acelerando la búsqueda a cambio de evaluar un subconjunto más limitado de opciones.

La validación cruzada se implementó utilizando el método de *k-fold cross-validation*, que permite una evaluación más robusta dividiendo el conjunto de datos en k subconjuntos y entrenando el modelo en k-1 subconjuntos, utilizando el subconjunto restante para la validación. Esta técnica es fundamental para reducir el riesgo de sobreajuste durante la optimización de hiperparámetros, asegurando que el modelo no solo aprenda de los

datos de entrenamiento, sino que también generalice adecuadamente sobre datos no vistos.

Para diagnosticar problemas de sobreajuste o subajuste, se utilizaron curvas de aprendizaje, que permiten visualizar cómo varía el rendimiento de los modelos en función del tamaño del conjunto de entrenamiento. Estas curvas proporcionaron información sobre el comportamiento del modelo conforme se aumentaba el número de muestras, ayudando a identificar cuándo un modelo aprende correctamente o memoriza los datos de entrenamiento.

Debido a que el problema presenta un desbalance en las clases, aunque se haya intentado corregir, se utilizó la métrica F1-score como criterio de evaluación. Esta métrica es especialmente útil en este tipo de problemas, ya que equilibra la precisión y la sensibilidad, siendo una mejor representación del rendimiento general.

## 5.7. Evaluación de modelos

El objetivo fue identificar el modelo que ofreciera el mejor equilibrio entre sensibilidad y precisión, priorizando la minimización de falsos negativos, un aspecto crítico en el contexto clínico de este proyecto.

Tal y como se ha especificado en el apartado Construcción de modelos de clasificación, se seleccionaron cinco algoritmos de clasificación ampliamente validados y usados en la literatura bioinformática.

Se utilizaron varias métricas para evaluar el rendimiento de los modelos: precisión, sensibilidad, *F1-Score*, AUC-ROC y *AUC Precision-Recall*. En este caso, no se ha utilizado la métrica de *accuracy*, ya que en la detección del cáncer los falsos negativos son mucho más graves que los falsos positivos, y la *accuracy* no refleja adecuadamente estos costos de error.

Para cada modelo, se generaron matrices de confusión que permitieron una visualización directa de la relación entre los verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Esto ayudó a comprender mejor los errores cometidos por los clasificadores.

También se generaron curvas *ROC* y *Precision-Recall*. Las curvas *ROC* facilitaron la evaluación del rendimiento del modelo, mientras que las curvas *Precision-Recall* fueron esenciales en el análisis de la relación entre precisión y sensibilidad, y evaluar el impacto de los falsos negativos.

Se evaluaron todos los modelos en base a su desempeño en el conjunto de datos de prueba.

## 5.8. Importancia de miARNs

La importancia de las características fue analizada para identificar qué miARNs resultaban más relevantes en la clasificación del cáncer. Se utilizaron los modelos *Random Forest* y *XGBoost*. Se emplearon dos enfoques específicos: la reducción del *índice de Gini* en el caso de *random forest* y la métrica *weight* en *XGBoost*. Estos dos parámetros son los predefinidos por ambos modelos.

En el modelo de *random forest*, que emplea el *índice de Gini* para la división de nodos, la importancia de las características se define en función de la cantidad de impureza que cada característica reduce en los árboles de decisión. Cuanto mayor es la reducción de impureza, mayor es la importancia de la característica.

Por su parte, *XGBoost* emplea una métrica llamada *weight*, que refleja el número de veces que una característica se selecciona para una división en los árboles de decisión. Cuanto más frecuente es la selección de un miARN para dividir un nodo, mayor será su importancia en el modelo.

Para realizar la evaluación de la importancia de los miARNs, se utilizaron los modelos previamente entrenados.

## 6. Resultados y discusión

### 6.1. Análisis exploratorio de miARNs

El análisis exploratorio de los datos reveló varios aspectos importantes relacionados con la estructura y distribución del *dataset* de expresión de miARN, los cuales se detallan a continuación:

#### *Estadísticas descriptivas*

Las estadísticas descriptivas básicas indicaron una considerable variabilidad en los niveles de expresión de los miARNs entre los experimentos. Las medias oscilaron entre 0,07 y más de 1000000, lo que sugiere la existencia de diferencias biológicas o técnicas entre los distintos estudios. La mediana, sin embargo, mostró valores de 0 en la mayoría de los casos, lo que refleja la predominancia de ceros en el *dataset*.

Se confirmó que no existían valores faltantes en ninguno de los experimentos, ya que todas las columnas del *dataset* contenían los 2419 registros correspondientes a los miARNs. Este resultado asegura la integridad de los datos y evita la necesidad de aplicar técnicas de imputación.

Se observó que al menos el 75% de los miARNs no presentaban niveles de expresión detectable en la mayoría de los experimentos, como se refleja en los percentiles del 25%, 50% y 75%, que resultaron ser 0 en todos los casos. Que los datos estén dispersos es un rasgo característico de los estudios de expresión génica y tendrá implicaciones en la selección de características y en los modelos de clasificación. La desviación estándar fue significativamente alta en relación con las medias, superando en algunos casos los 20000. Esto pone de manifiesto la existencia de una gran variabilidad en la expresión de los miARNs. Además, los valores máximos detectados en algunos experimentos superaron el millón, lo que indica la presencia de miARNs con una expresión extremadamente elevada en un pequeño subconjunto de experimentos. Estos podrían ser considerados como valores atípicos.

Dada la disparidad observada entre los valores de expresión, se identificó la necesidad de aplicar técnicas de normalización para reducir la asimetría de los datos y evitar que los valores atípicos distorsionen los modelos de clasificación. En este sentido, se valoró la implementación de una transformación logarítmica para corregir el sesgo en la distribución de los datos; así como normalizaciones específicas para datos de miARNs como TMM. Además, se valoró oportuno eliminar aquellos miARNs que no presentaban expresión significativa en la mayoría de los experimentos.

Las estadísticas se detallan en el archivo stats\_mirna\_raw.csv, que se encuentra dentro del repositorio de GitHub. La ruta completa es: results/statistics/stats\_mirna\_raw.csv.

### Distribución de datos

Los histogramas generados para las 30 muestras aleatorias mostraron una clara tendencia hacia una distribución sesgada a la derecha, con un pico alrededor de 0. Los datos estaban muy concentrados en valores bajos, con una larga cola hacia la derecha (Figura 6). Esto indicó que, en la mayoría de los casos, los miARNs presentaban niveles de expresión muy bajos o nulos en las muestras, lo que es característico de los estudios de expresión génica de miARNs. Además, confirmó las conclusiones obtenidas en las estadísticas descriptivas.

La distribución altamente sesgada, concentrando la mayoría de los datos en un rango muy pequeño, hubiera dificultado la comparación entre muestras; por lo que se decidió aplicar una transformación logarítmica ( $\log_2(x + 1)$ ) a los datos de *read counts*. El uso de  $\log_2$  es habitual en estudios de expresión génica, como los de miARN, ya que facilita la comparabilidad de resultados con otros estudios similares (Pant et al., 2019; Sarkar et al., 2021).

Tras aplicar la transformación logarítmica, se observó una reducción en la asimetría de las distribuciones, suavizando el sesgo hacia la derecha y moderando los valores extremos (Figura 7). Sin embargo, a pesar de esta normalización, las distribuciones siguieron presentando un sesgo considerable, confirmando que gran parte de los miARNs tienen baja o nula expresión en muchas de las muestras. La prueba de Shapiro-Wilk también confirmó que los datos de expresión de los miARNs no seguían una distribución normal, dado que el p-valor obtenido en la prueba es mucho menor que el nivel de significancia establecido (0,05). Este resultado enfatizó la importancia de emplear técnicas de normalización robustas o modelos de clasificación que no dependan de esta suposición.

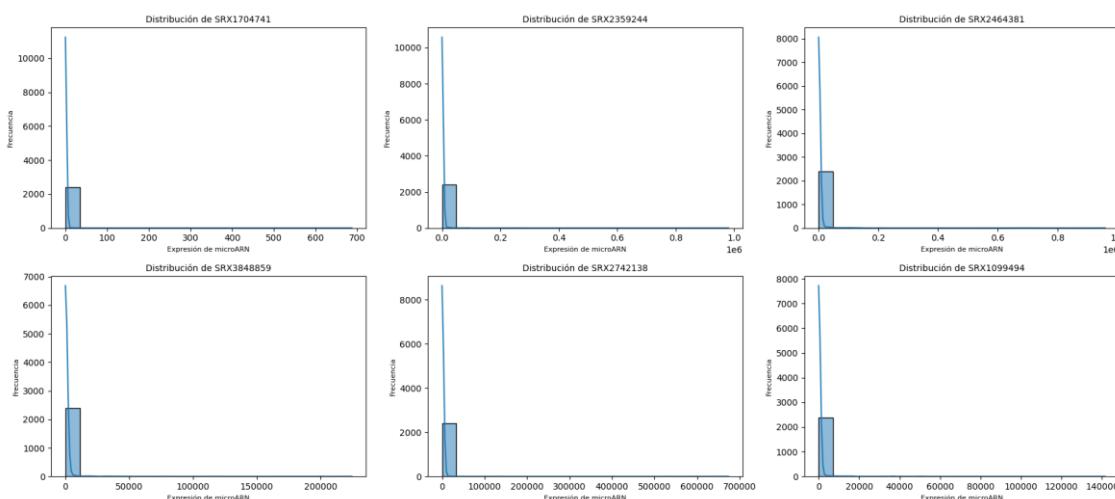


Figura 6: Histogramas antes de la normalización.

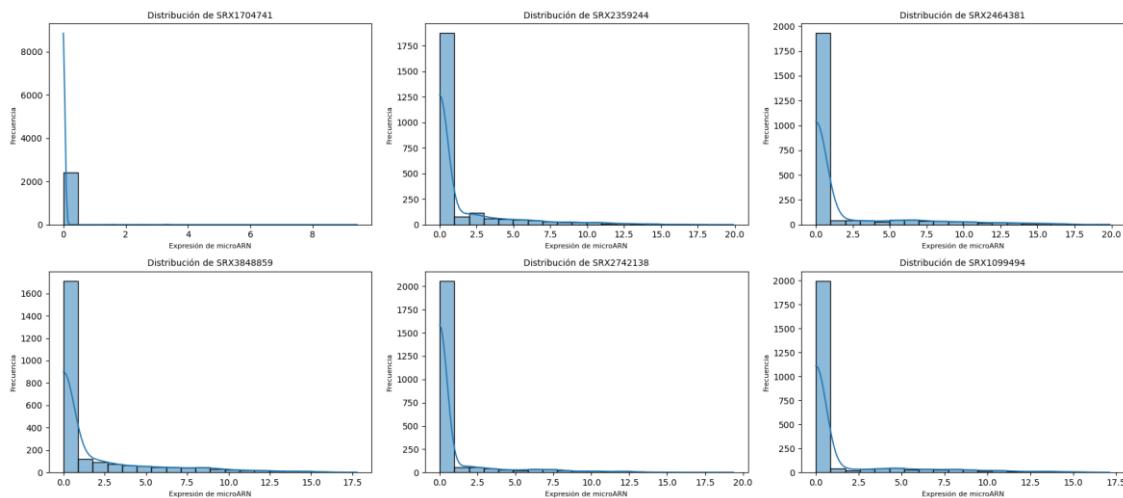


Figura 7: Histogramas después de la normalización.

#### Detección de valores atípicos

La aplicación del método IQR reveló que 1556 de las 1559 muestras contenían valores que fueron clasificados como atípicos. Sin embargo, en este contexto, es importante destacar que la presencia de *outliers* no necesariamente indica errores experimentales o de muestreo (X. Chen et al., 2020). Dado que los niveles de expresión de los miARNs pueden variar considerablemente entre diferentes muestras biológicas, los valores atípicos identificados podrían representar variaciones biológicas reales. De hecho, la dispersión observada de los *outliers* a lo largo de casi todas las muestras refuerza la idea de que muchas de estas variaciones pueden ser inherentes a la biología del sistema analizado, más que a errores técnicos.

Los diagramas de caja confirmaron la existencia de múltiples valores atípicos en prácticamente todas las muestras del *dataset*, como se observa en los gráficos generados (Figura 8). Estos valores atípicos son consistentes con la variabilidad esperada en estudios de expresión génica de miARNs. No se detectó una concentración significativa de *outliers* en ninguna muestra específica, lo que sugiere que las diferencias observadas están distribuidas de manera uniforme a lo largo de las muestras. Por lo tanto, aunque los *outliers* estuvieran presentes en casi todas las muestras, no se descartaron automáticamente, ya que podrían representar fenómenos biológicos relevantes.

El tratamiento de estos valores se abordará más adelante en la fase de preprocesamiento, donde se determinará el método más adecuado.

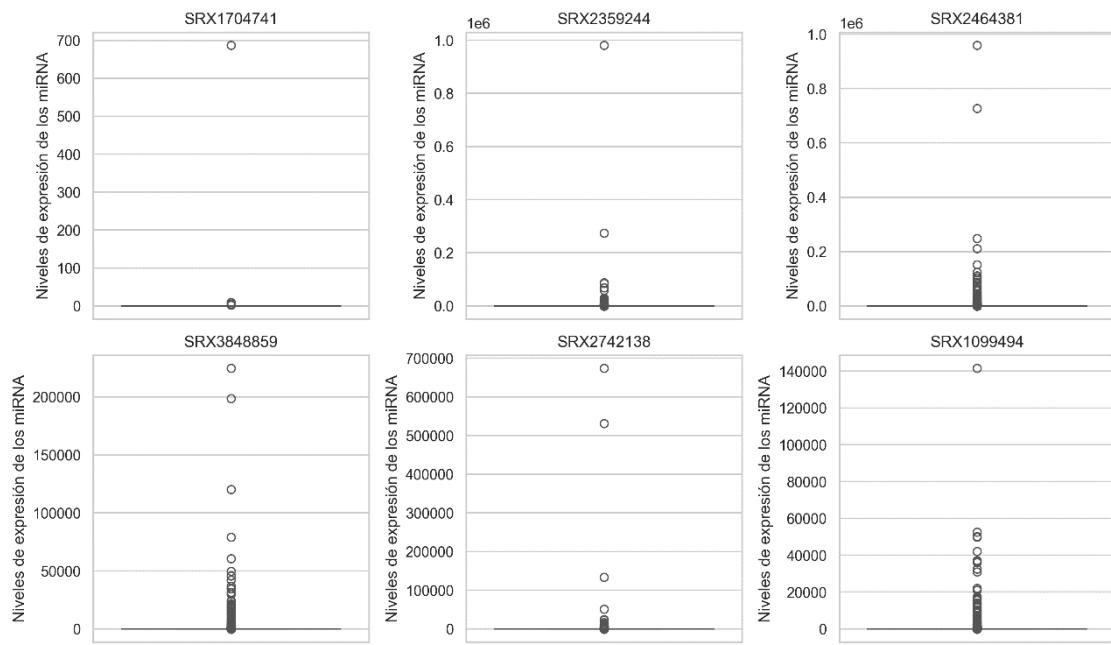


Figura 8: Diagramas de caja de diferentes experimentos.

#### Análisis de correlación

Tras la transposición del *dataset*, se detectaron 15719 pares de miARNs con una correlación absoluta superior a 0,7 y 5403 pares con una correlación absoluta superior a 0,85. Estos resultados evidencian una correlación elevada entre muchos miARNs, lo que sugiere redundancia en los datos. Además, una alta correlación entre miARNs puede generar multicolinealidad, lo que afecta la estabilidad y la interpretabilidad de algunos modelos de clasificación. Es importante señalar que tal redundancia podría ser indicativa de coexpresión, donde ciertos miARNs podrían estar regulando o participando en procesos biológicos similares (Hausser & Zavolan, 2014).

El mapa de calor de las 25 correlaciones más altas (Figura 9) muestra grupos de miARNs altamente correlacionados. Esto sugiere que la eliminación o combinación de ciertos miARNs, mediante métodos de selección de características o reducción de dimensionalidad, podría ser una estrategia adecuada para evitar problemas de multicolinealidad que pudieran afectar a la precisión de los modelos de clasificación.

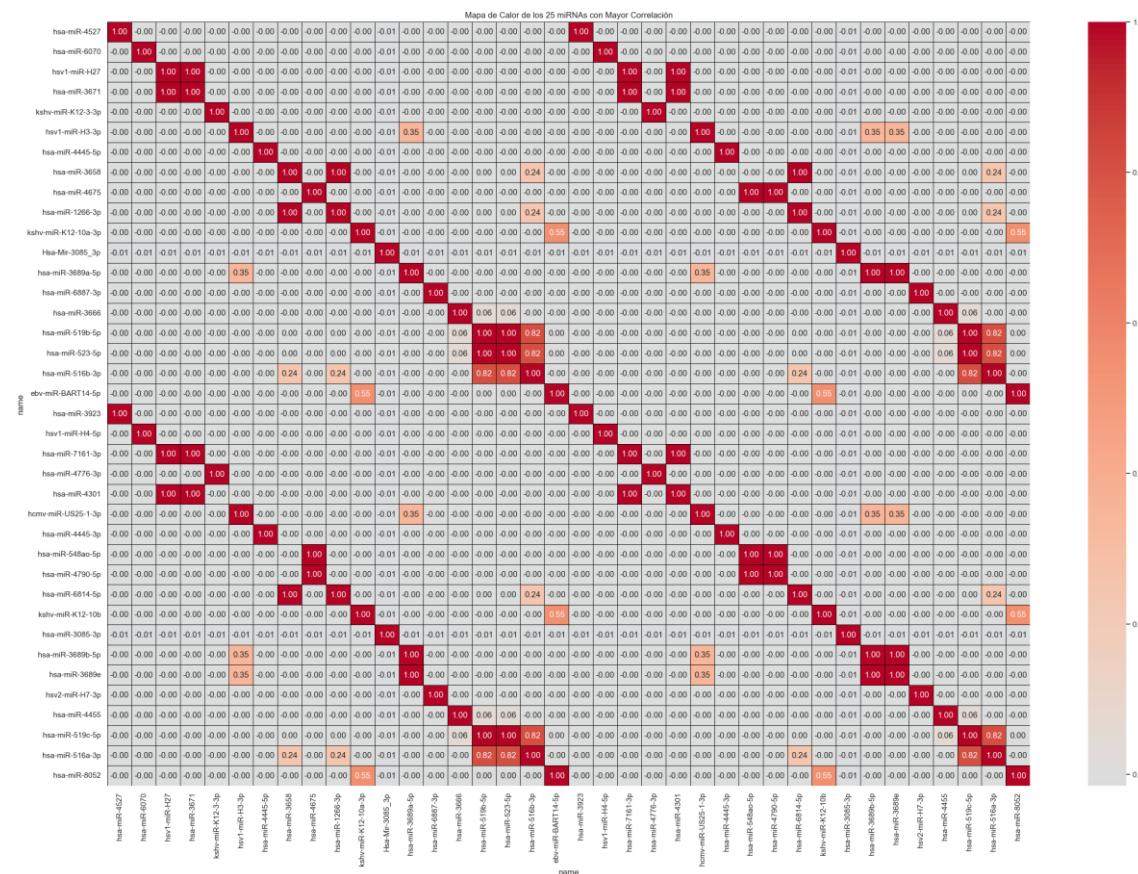


Figura 9: Mapa de calor de los 25 miARNs con mayor correlación absoluta

En cuanto a la correlación entre los miARNs y la variable objetivo **Cáncer**, se observaron valores que fluctúan entre 0,62 y 0,41 en términos absolutos para las 40 correlaciones más altas (Figura 10). Estos resultados indican que ciertos miARNs tienen una relación significativa con el estado de cáncer de las muestras, lo que sugiere que podrían ser útiles como biomarcadores para distinguir entre pacientes con cáncer y aquellos sin la enfermedad.

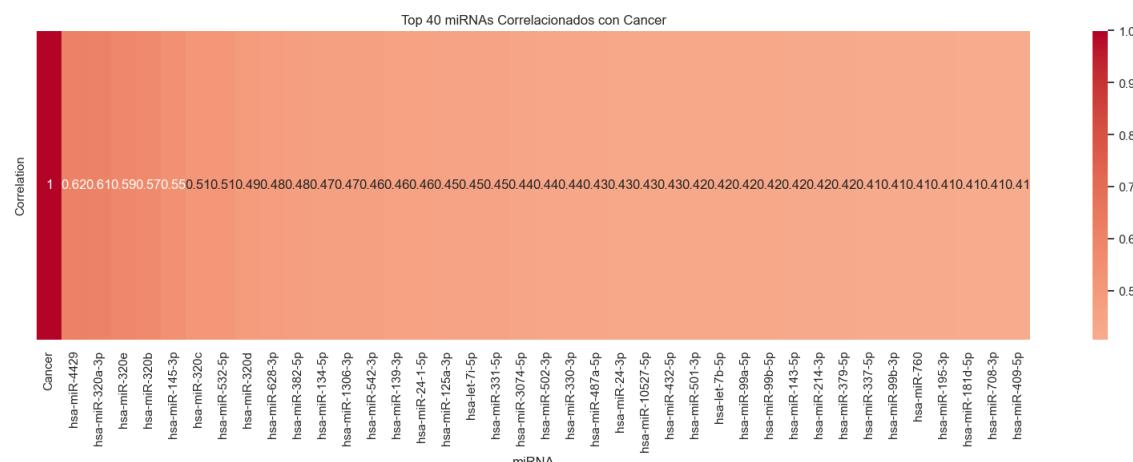


Figura 10: Top 40 miARNs correlacionados con la variable cáncer.

### Análisis multivariado

En el análisis de PCA entre muestras no se observaron patrones muy claros, aunque se pueden diferenciar dos cuernos, uno por cada componente principal (Figura 11). Algunas muestras se alejan de los grupos principales, lo que indica la presencia de valores atípicos. Estos puntos podrían representar muestras contaminadas o con características biológicas únicas, lo que merece una inspección adicional. La primera componente principal (PC1) explicó el 80,19% de la variabilidad en los datos, mientras que la segunda componente (PC2) explicó el 8,77%. Esto sugiere que gran parte de la estructura de los datos puede entenderse observando solo estos dos componentes.

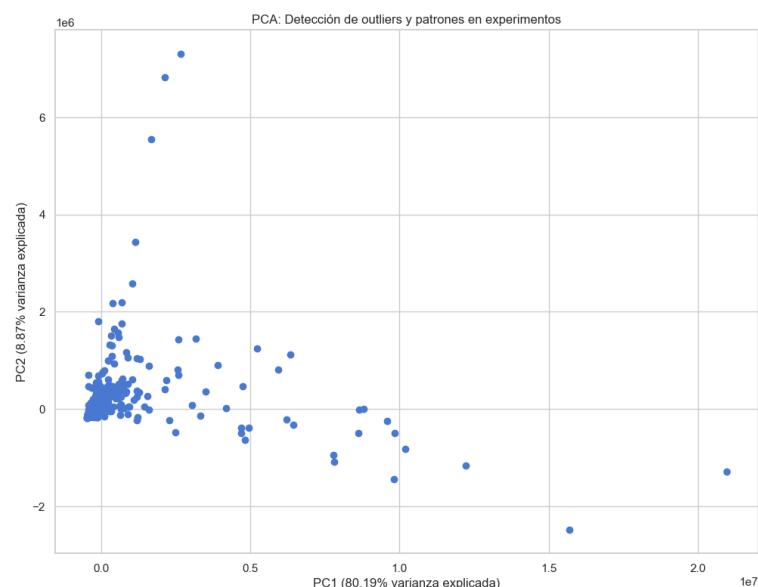


Figura 11: Análisis PCA de los experimentos.

Los gráficos de dispersión realizados con las variables *Healthy* y *Cancer* también permitieron visualizar un claro agrupamiento de las muestras según la condición de salud: las muestras de pacientes con cáncer y las muestras saludables formaron grupos distintos (Figura 12). Esto sugiere que los niveles de expresión de miARN capturados en los primeros componentes principales son útiles para diferenciar entre clases, apoyando la hipótesis de que los miARNs podrían ser buenos biomarcadores para distinguir entre cáncer y estados saludables. En cuanto a la variable *Fluid*, las muestras de *whole blood* mostraron un comportamiento distintivo, dispersándose más en el PC1 (Figura 13). Esto sugiere que la sangre total tiene un patrón de expresión de miARN que lo diferencia claramente de otros tipos de fluido, como plasma, sérum o células sanguíneas. Este hallazgo es particularmente interesante desde un punto de vista biológico y sugiere que las muestras de sangre total podrían ser las más útiles para futuros análisis de clasificación.

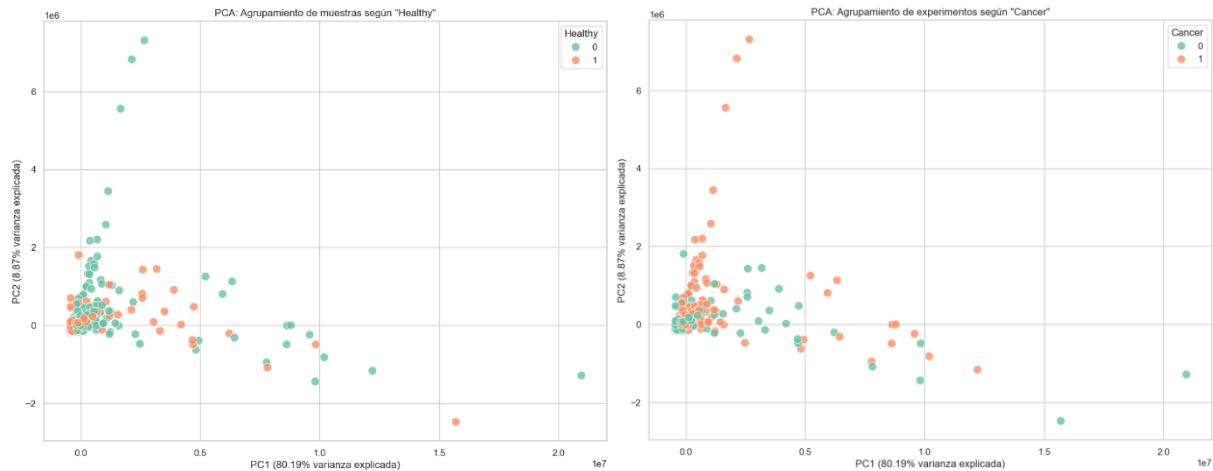


Figura 12: Healthy y Cancer - Gráfico de dispersión.

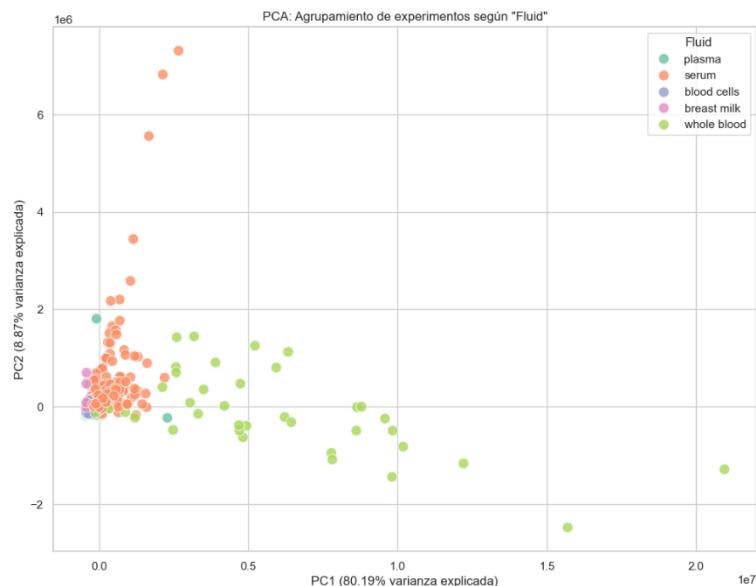


Figura 13: Fluid - Gráfico de dispersión.

El análisis *clustering* con *k-means* confirmó la separación observada en los gráficos de PCA. Para  $k=2$ , las muestras se agruparon claramente en dos clústeres, coincidiendo con los cuernos observados en los gráficos de dispersión (Figura 14). Con  $k=5$ , aunque las muestras no se agruparon de manera tan clara, se identificó un pico correspondiente a las muestras de pacientes con cáncer, lo que refuerza la hipótesis de que los niveles de miARN permiten distinguir entre pacientes con y sin cáncer (Figura 14). El método del codo sugirió que  $k=5$  era el número óptimo de clústeres, lo que también capturó las diferencias entre los tipos de fluidos de manera adecuada. Además, este método dio como resultado el mismo dígito que número de fluidos tras eliminar los valores nulos del dataset integrado.

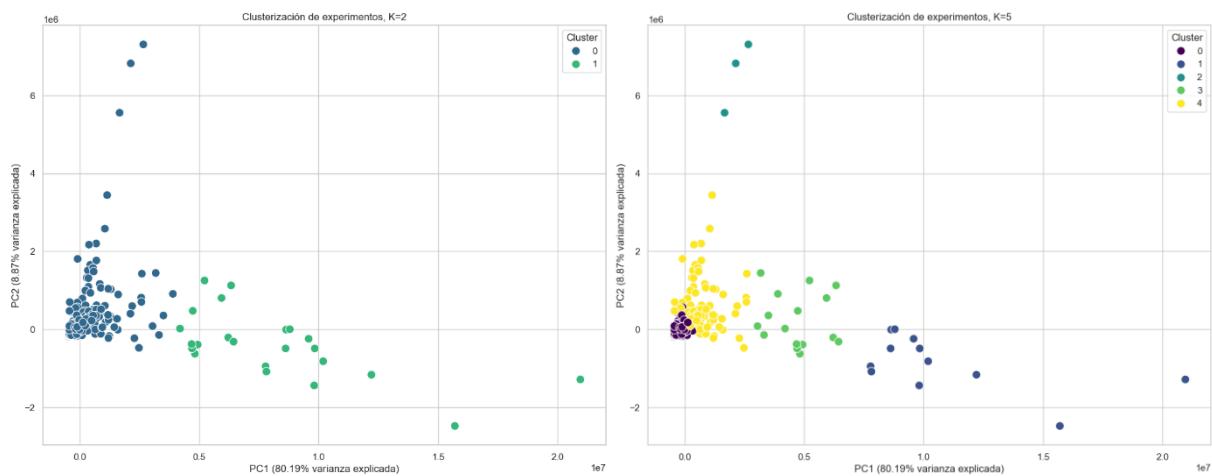


Figura 14: Análisis de clustering.

El análisis de PCA entre miARNs reveló que ciertos miARNs se agrupan, lo que sugiere que podrían estar co-regulados o cumplir funciones similares. Solo tres miARNs se encontraron fuera del agrupamiento principal, lo que podría indicar que estos tienen patrones de expresión únicos o poco frecuentes, o que son valores atípicos. La primera componente principal explicó el 77,22% de la variabilidad entre los miARNs, mientras que la segunda explicó el 10,85% (Figura 15).

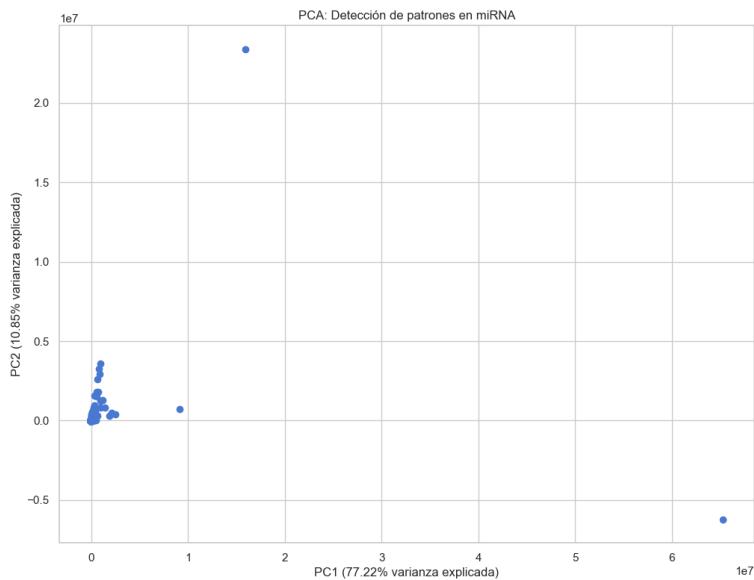


Figura 15: Análisis PCA de miARNs.

## 6.2. Análisis exploratorio de *metadata*

Mediante el resumen estadístico, se identificó que los metadatos *Sex*, *Fluid*, *Extraction*, *Library*, *Healthy*, *Cancer* y *Desc*, tenían valores faltantes. Estos requerirán un manejo adecuado para asegurar la integridad de los análisis posteriores. Los pasos realizados se detallan en el apartado Preprocesamiento.

Gracias a la Figura 16 se detectó un desbalance en la característica *Cancer*, que será necesario corregir en etapas posteriores del análisis mediante técnicas de balanceo de clases, como *oversampling* o *undersampling*.

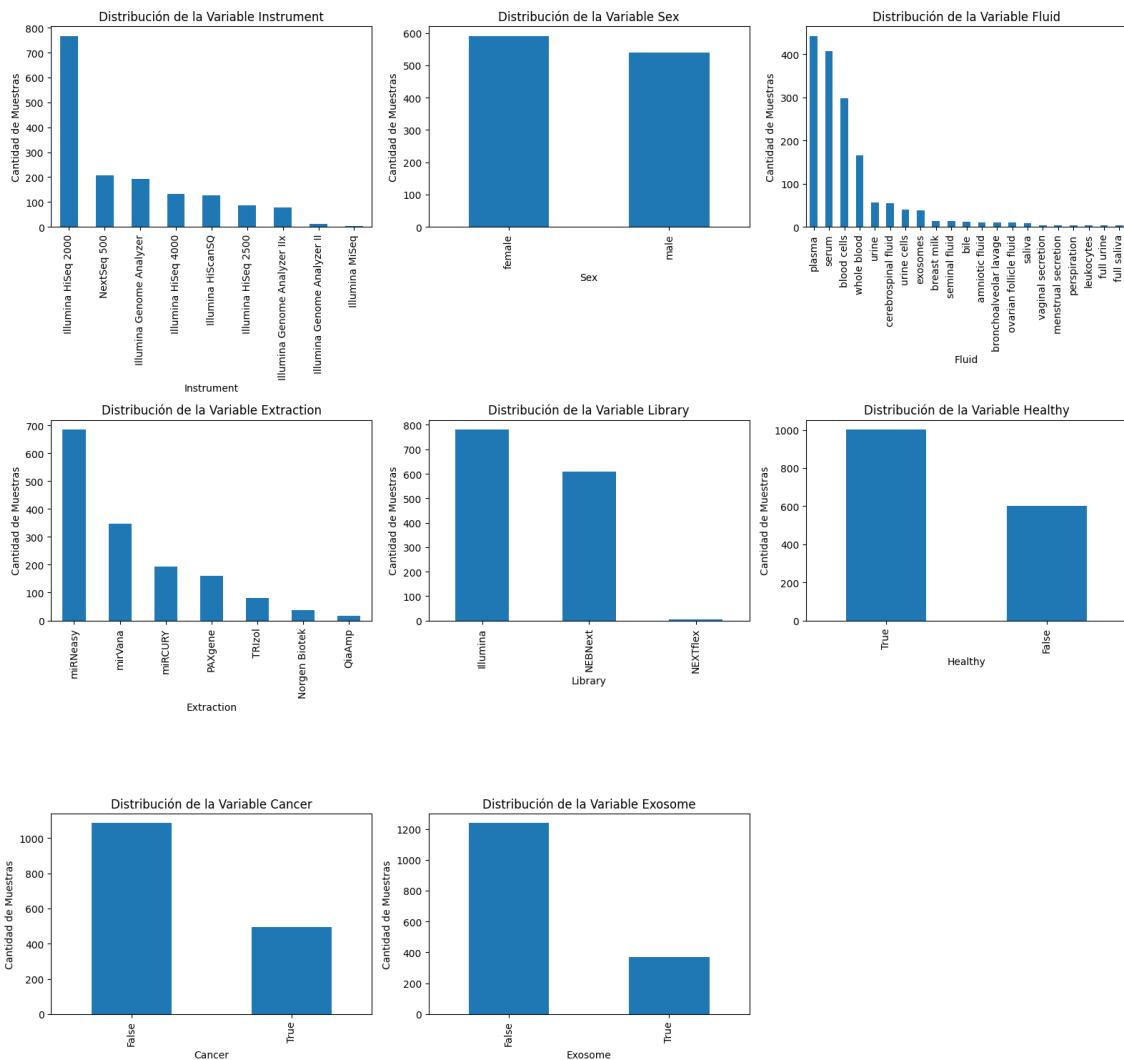


Figura 16: Distribución de las categorías de la metadata.

### 6.3. Preprocesamiento de miARNs y metadata

#### 6.3.1. Filtrado de miARNs con baja expresión

Tras aplicar el filtro, se eliminó una cantidad considerable de miARNs que tenían recuentos de expresión iguales a cero en al menos el 60% de las muestras. Este umbral se estableció para asegurar que los miARNs que solo estaban presentes en una pequeña fracción de las muestras, o que no aportaban información significativa, fueran excluidos del análisis.

Inicialmente, el *dataset* contenía 2419 miARNs, y después del filtrado, se retuvieron 415 miARNs, lo que representa una reducción del 82,84% del total de características originales. Este paso permitió concentrarse en los miARNs con mayor relevancia biológica, eliminando aquellos que probablemente se hubieran comportado como ruido en el análisis.

### 6.3.2. Normalización de datos mediante TMM

Tal y como se ha explicado en el apartado Preprocesamiento, se utilizó TMM para normalizar los *read counts*. El uso de TMM es recomendable ya que garantiza una comparabilidad precisa entre las muestras. Asimismo, como ya se verificó en el Análisis exploratorio de miARNs, la mayoría de los miARNs presentaban un alto número de valores iguales a cero en muchas muestras, lo que indica que muchos de ellos están expresados en niveles extremadamente bajos o no están expresados en la mayoría de las muestras. En este contexto, TMM es particularmente útil, ya que permite normalizar los datos teniendo en cuenta estas expresiones extremas, ajustando adecuadamente las diferencias en la composición de las muestras.

Existen otros métodos de normalización, como CPM y RPM, que ajustan por la profundidad de secuenciación (tamaño de la biblioteca), o RPKM/FPKM y TPM, que también ajustan función de la longitud del gen. Estos métodos son útiles para comparar la expresión dentro de una misma muestra. Sin embargo, en el caso de este proyecto, es imprescindible poder comparar entre muestras, por lo que no sirven.

Después de aplicar la normalización mediante TMM, el número de *outliers* presentes en las muestras disminuyó significativamente. Antes de la normalización, el análisis descriptivo había revelado que prácticamente todas las columnas (miARNs) contenían valores extremos, lo que indicaba una alta variabilidad en la expresión entre las muestras. Tras la normalización, se observó una mejora en la distribución de los datos, con una reducción de *outliers* por muestra, lo que sugiere que el proceso de normalización fue exitoso en reducir el sesgo.

A pesar de que las distribuciones seguían siendo asimétricas, los histogramas posteriores a la normalización mostraron una mejora en la dispersión de las expresiones, lo que se reflejó también en el suavizado de los gráficos KDE (Kernel Density Estimation). Aunque la variabilidad entre las muestras aún era notable, se concluyó que los resultados de la normalización habían mejorado la comparabilidad de las expresiones entre las muestras (Figura 17). Sin embargo, debido al sesgo residual observado en las distribuciones, se determinó que una transformación logarítmica adicional sería necesaria para optimizar el análisis posterior, mejorando aún más la homogeneidad entre las muestras.

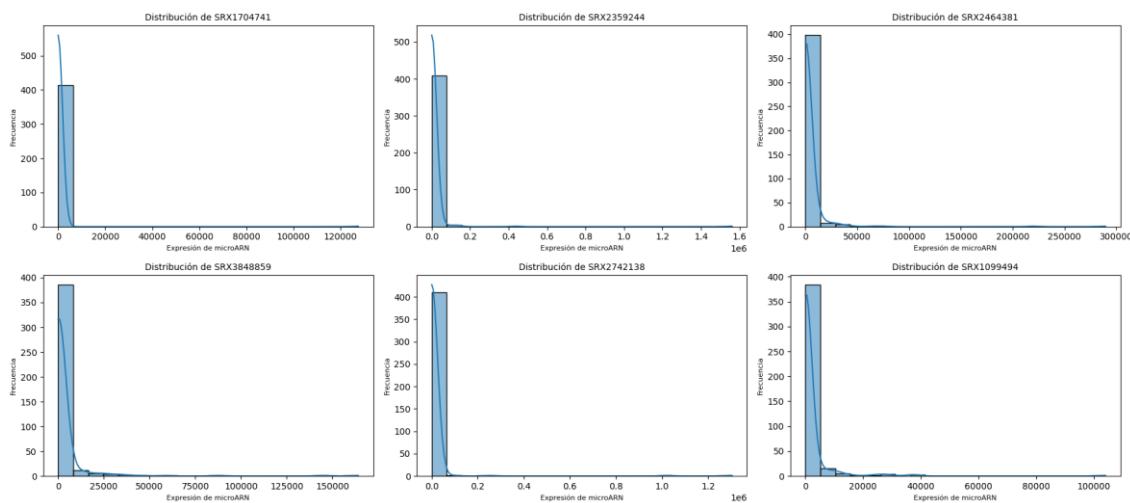


Figura 17: Histogramas después de la normalización con TMM.

### 6.3.3. Estandarización de datos con $\log_2(x + 1)$

La transformación  $\log_2(x + 1)$  fue elegida por diversas razones. Robinson & Oshlack (2010) explican varias razones por las cuales es apropiado. En primer lugar, los cambios logarítmicos son una manera estándar de medir variaciones en la expresión génica, ya que convierten las distancias entre los valores de expresión en una escala más apropiada para el análisis. En segundo lugar, a pesar de la normalización TMM, los datos originales de miARN seguían mostrando una distribución altamente asimétrica, con un pequeño número de miARNs con expresiones muy elevadas y un gran número con expresiones bajas o nulas, lo que generaba un sesgo que podría comprometer la fiabilidad de los análisis. La transformación logarítmica ayuda a mitigar esta asimetría, acercando los datos a una distribución más normal, lo cual es necesario para muchos análisis estadísticos. Además, al aplicar esta transformación se atenúan los valores extremos de algunos miARNs que podrían dominar los análisis y ocultar patrones importantes. Esto suaviza el impacto de los valores atípicos, haciendo los datos más comparables entre las muestras. Por último, la transformación logarítmica estabiliza la varianza de los datos, reduciendo la posibilidad de obtener falsos positivos o negativos en estudios de expresión diferencia.

Después de aplicar la transformación  $\log_2(x + 1)$  al dataset normalizado con TMM, se observó una clara mejora en la distribución de los datos de expresión de miARN. Los histogramas mostraron que las distribuciones se transformaron hacia una forma más simétrica y centrada (Figura 18). La reducción de la asimetría indica que los valores de expresión de miARN están ahora más balanceados y comparables entre muestras. Esto facilitará la detección de patrones significativos y relaciones biológicas relevantes en el análisis posterior.

Además, esta transformación contribuyó a que la varianza en los datos se estabilizara y se ajustara mejor a las suposiciones de normalidad que subyacen en varios modelos de clasificación y métodos estadísticos.

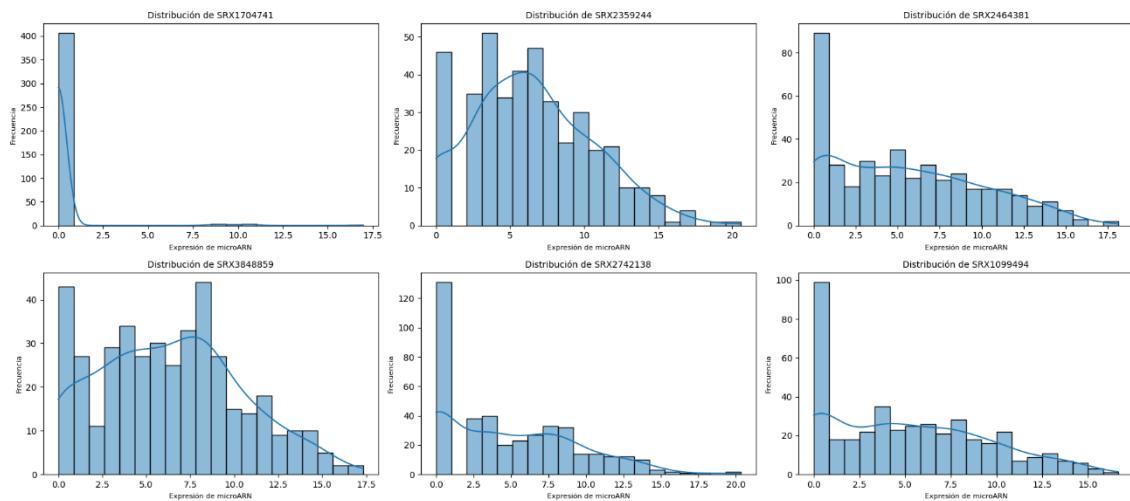


Figura 18: Histogramas después de la normalización con  $\log_2(x + 1)$ .

### 6.3.4. Tratamiento de valores faltantes en metadata

Se detectaron valores faltantes en las columnas *Sex*, *Fluid*, *Extraction*, *Library*, *Healthy* y *Desc*, con porcentajes variables entre ambos datasets (Tabla 1). En ambos casos (entrenamiento y prueba) se observan patrones que indicaban que la ausencia de datos está relacionada con observaciones específicas (Figura 19).

Tabla 1: Valores faltantes en mirna y metadata.

	Entrenamiento		Prueba	
	Nº valores faltantes	Porcentaje de valores faltantes	Nº valores faltantes	Porcentaje de valores faltantes
<b>Sex</b>	366	29,95%	95	31,05%
<b>Fluid</b>	2	0,16%	1	0,33%
<b>Extraction</b>	74	6,06%	14	4,58%
<b>Library</b>	174	14,24%	35	11,44%
<b>Healthy</b>	2	0,16%	1	0,33%
<b>Desc</b>	435	35,60%	110	39,95%

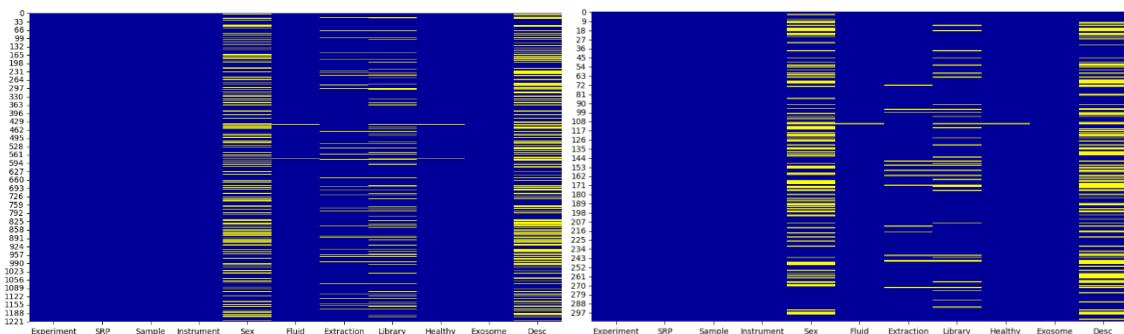


Figura 19: Mapa de calor para verificar valores faltantes. El gráfico de la derecha corresponde al conjunto de datos de entrenamiento y el de la izquierda al de prueba.

Para abordar los valores faltantes, se aplicaron diferentes estrategias según la importancia y la cantidad de datos ausentes en cada columna. En el caso de la variable *Sex*, al ser crucial para el análisis, se imputaron los valores nulos mediante el método KNN. Para las columnas *Fluid* y *Healthy*, dado el bajo porcentaje de valores faltantes, se optó por eliminar las observaciones incompletas. En cuanto a *Extraction* y *Library*, la imputación no fue considerada necesaria porque no influyen directamente en las características biológicas. Por último, la columna *Desc* fue eliminada debido a su alto porcentaje de datos faltantes y su baja relevancia para el estudio.

Tal y como explica Troyanskaya et al. (2001), el uso de KNN para la imputación de valores faltantes es un enfoque altamente utilizado en bioinformática siempre y cuando se cumplan ciertos factores (Hastie et al., 2009): Como la imputación se realiza utilizando los valores de las observaciones más similares en el conjunto de datos, este método funciona si existe una correlación entre los datos. Además, KNN depende de las distancias entre observaciones, por lo que los datos deben estar escalados adecuadamente. Por último, el algoritmo funciona mejor si las observaciones están distribuidas densamente y de manera homogénea en el espacio de características. En el caso de los datos de este trabajo, como se analizó en el Análisis exploratorio de miARNs, los datos están correlacionados y también fueron escalados de manera adecuada con  $\log_2(x + 1)$ .

El uso del método KNNImputer fue efectivo para llenar los valores faltantes en la columna *Sex*. La imputación basada en observaciones similares permitió mantener la homogeneidad de las características y evitar la pérdida de información importante. Además, al eliminar la columna *Desc*, que presentaba un porcentaje excesivo de datos faltantes y no era esencial para el análisis, se simplificó el *dataset* sin comprometer su utilidad.

El análisis de *outliers* previo a la imputación evitó distorsiones en el proceso de KNN, asegurando que las imputaciones fueran precisas y representativas de los patrones subyacentes en los datos. Este paso, que se explica en el siguiente punto, fue clave para garantizar que las observaciones imputadas fueran consistentes con las características biológicas de las muestras.

### 6.3.5. Análisis de valores atípicos

En la visualización PCA se observó que las muestras de cáncer tendían a agruparse en dos grupos distintos, lo que proporcionó una referencia visual útil para evaluar la pertinencia de los *outliers* detectados (Figura 20).

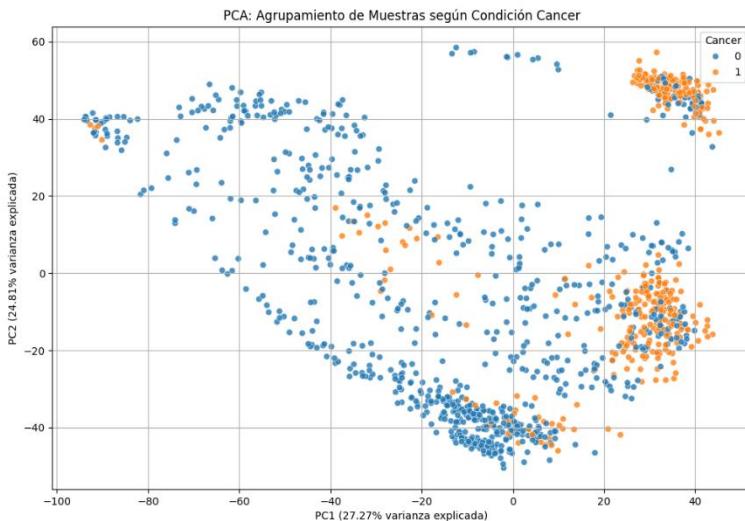


Figura 20: Valores atípicos en miARN. Los puntos naranjas corresponden a muestras de cáncer.

Tras aplicar los métodos descritos en el apartado Valores atípicos en miARNs, se obtuvieron los siguientes resultados:

#### *Isolation Forest*

Hiperparámetros: “n\_estimators = 300” y “contamination = 0.05”.

Se detectaron 61 *outliers*. Esta configuración fue elegida por su equilibrio entre la detección de *outliers* sin sobreajustar el modelo.

#### *LOF*

Hiperparámetros: “n\_neighbors = 10” y “contamination = ‘auto’”.

Se detectaron 41 *outliers*. Esta configuración se seleccionó por su capacidad de detectar *outliers* en regiones de baja densidad de datos sin ser demasiado conservadora.

#### *DBSCAN*

DBSCAN no fue empleado en este análisis debido a la naturaleza de los datos. Éstos no presentaban agrupaciones muy claras en el espacio de características, por lo que el algoritmo detectó un número altísimo de valores atípicos.

#### *Distancias Multivariadas (Mahalanobis)*

Configuración: Un umbral del percentil 97,5%

Se identificaron 31 *outliers*. Estos *outliers* presentaban patrones coherentes en el análisis PCA y contribuyeron de manera correcta a la selección.

Los valores atípicos seleccionados fueron aquellos que coincidían en al menos dos de los tres métodos principales. De esta manera, se identificaron 17 valores atípicos finales que se consideraron problemáticos y se eliminaron del conjunto de entrenamiento (Figura 21).

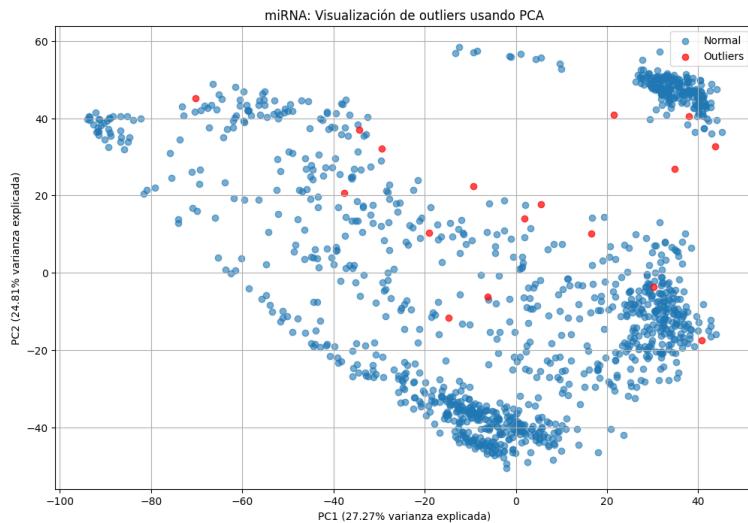


Figura 21: Valores atípicos en miARNs. Los puntos rojos corresponden a las muestras eliminadas.

La integración de los *mirna* y *metadata* permitió una identificación extra y más precisa de los *outliers*. En este caso, estos fueron los resultados obtenidos:

#### *Isolation Forest*

Hiperparámetros: "n\_estimators = 300" y "contamination = 'auto'".

Se detectaron 172 *outliers*.

#### *LOF*

Hiperparámetros: "n\_neighbors = 10" y "contamination = 'auto'".

Se detectaron 32 *outliers*.

El uso de *Isolation Forest* y LOF produjo diferentes cantidades valores atípicos, pero la combinación de ambas técnicas proporcionó un conjunto más robusto. De los valores atípicos detectados, solo 9 valores atípicos coincidieron en ambos métodos (Figura 22), lo que asegura que solo se eliminaron aquellos valores que, en múltiples enfoques, se consideraron anómalos.

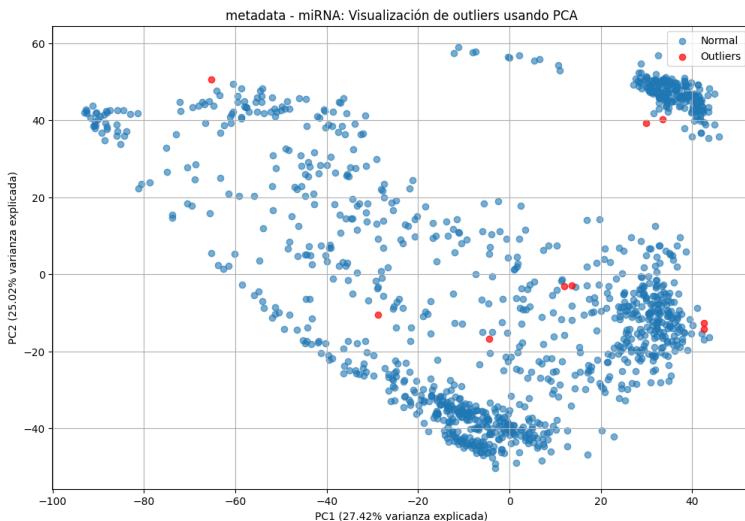


Figura 22: Valores atípicos en miARNs junto con metadata. Los puntos rojos corresponden a las muestras eliminadas.

En los gráficos PCA correspondientes a los valores atípicos detectados se visualiza que los puntos seleccionados como *outliers* no corresponden a clústeres densos, validando su eliminación. Esto resultó en un conjunto de datos más limpio y equilibrado, mejor preparado para entrenar modelos de clasificación robustos que sean capaces de generalizar a nuevos datos, incluyendo aquellos con perfiles de expresión atípicos.

El análisis detallado se puede revisar en el apartado Apéndice II: Detección de valores atípicos.

### 6.3.6. Balanceo de clases

La aplicación de SMOTE al conjunto de entrenamiento dio como resultado una mejora en la distribución de las clases. Antes del balanceo, el *dataset* presentaba un notable desbalance entre la clase mayoritaria y la clase minoritaria de *Cancer*, lo que hubiera limitado la capacidad del modelo de detectar correctamente instancias de la clase minoritaria. Esto sucede porque los modelos tienden a sesgar los modelos hacia la clase predominante, afectando la capacidad del modelo para detectar correctamente la clase minoritaria. Tras aplicar SMOTE, el número de observaciones en el conjunto de entrenamiento se incrementó significativamente, alcanzando un total de 1606 muestras balanceadas, lo que permitió una representación equitativa de ambas clases para mejorar la capacidad del modelo de clasificación.

## 6.4. Análisis exploratorio de datos posprocesamiento

En esta sección se presentan los resultados obtenidos tras realizar el análisis EDA posprocesamiento. El propósito de este análisis fue garantizar la calidad de los datos y evaluar su estado antes de seleccionar las características y construir los modelos de clasificación. A continuación, se detallan los principales hallazgos:

### *Revisión de los tipos de datos*

Los tipos de datos asignados a cada variable fueron correctos, sin inconsistencias detectadas. Las variables categóricas y numéricas se mantuvieron adecuadamente diferenciadas.

### *Resumen estadístico*

El análisis estadístico de las variables numéricas reveló que los miARNs mostraban una amplia gama de valores de expresión promedio. Algunos miARNs como hsa-miR-486-5p (media = 14,00) y hsa-let-7i-5p (media = 13,29) tenían niveles de expresión considerablemente más altos, mientras que otros como hsa-miR-376a-3p (media = 1,87) y hsa-miR-1304-5p (media = 1,71) tenían niveles de expresión mucho más bajos. También diferían en la variabilidad; pero es mucho menor que en los datos sin procesar. Asimismo, la diferencia entre los valores mínimos (frecuentemente 0) y los máximos altos sugirió que muchos miARNs tenían una distribución sesgada. Por último, los valores en el percentil 25 y la mediana indicaron que muchos miARNs tienen niveles de expresión nulos o cercanos a cero en al menos el 50% de las muestras; sin embargo, el percentil 75 sí que tuvo valores diferentes a 0.

Las estadísticas se detallan en el archivo stats\_mirna\_processed.csv, que se encuentra dentro del repositorio de GitHub. La ruta completa es: results/statistics/stats\_mirna\_processed.csv.

### *Inspección de valores faltantes:*

No se encontraron valores faltantes.

### *Distribución de los datos*

Los histogramas y gráficos KDE mostraron que la mayoría de las variables seguían una distribución sesgada a la derecha. Sin embargo, la asimetría se había reducido de manera notoria respecto a los datos sin procesar. Más aún, en estos gráficos se puede observar un segundo pico pronunciado en casi todos los miARNs; por lo que, si se eliminases los niveles de expresión cercanos a 0, se obtendrían histogramas con una distribución cercana a la normal (Figura 23).

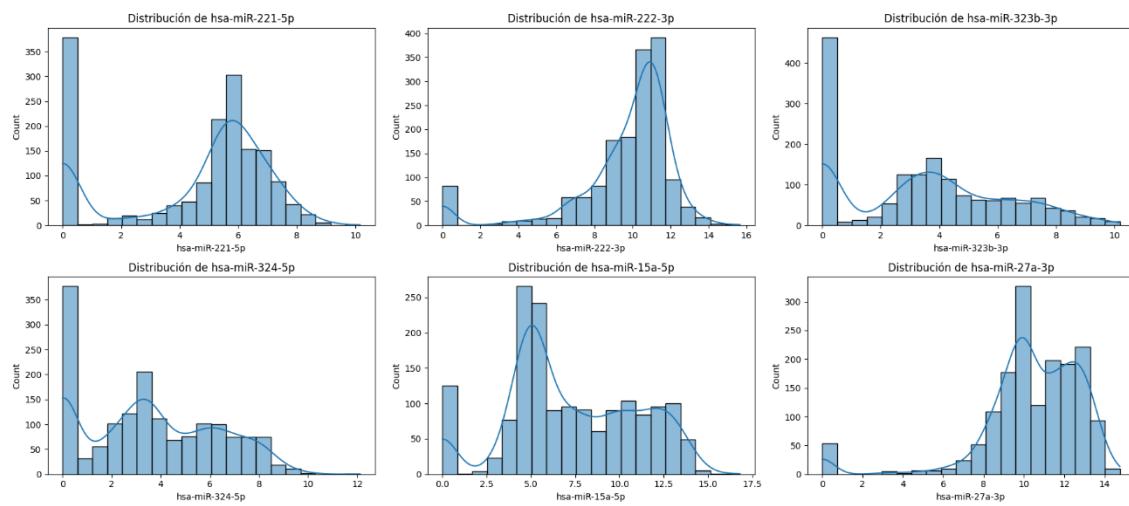


Figura 23: Histogramas tras el preprocesamiento de datos.

#### Detección de valores atípicos

Los diagramas de caja mostraron que el número de valores atípicos se había reducido. Además, el rango intercuartílico también se había reducido en todos los miARNs (Figura 24).

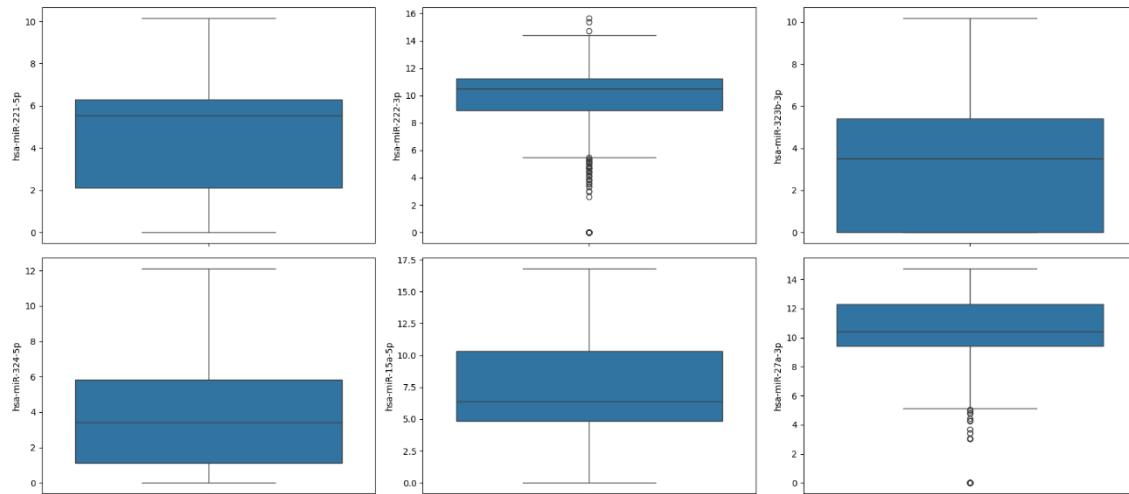


Figura 24: Diagramas de caja tras el preprocesamiento de datos.

#### Correlación de características

##### Correlación entre miARN:

El mapa de calor de la matriz de correlación reveló que un gran número de variables de expresión de miARN presentaban correlaciones fuertes (Figura 25). A partir de estos resultados se acentúa la necesidad de seleccionar miARNs para reducir el riesgo de colinealidad.

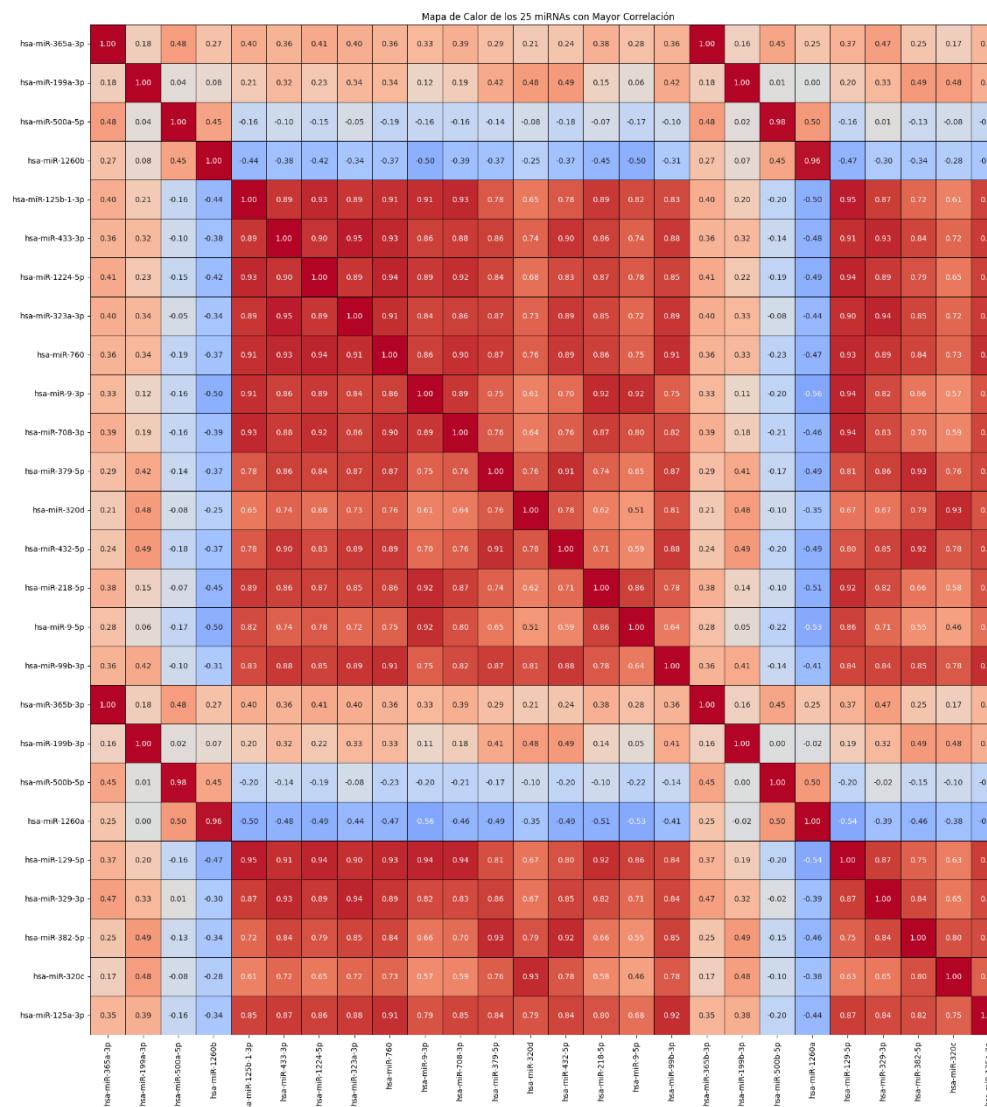


Figura 25: Mapa de calor con correlaciones entre miARNs.

### Correlación entre miARNs y cáncer

Se observaron correlaciones significativas entre ciertos miARNs y la presencia de cáncer, lo que sugiere que estos miARNs son potencialmente útiles para la clasificación (Figura 26).

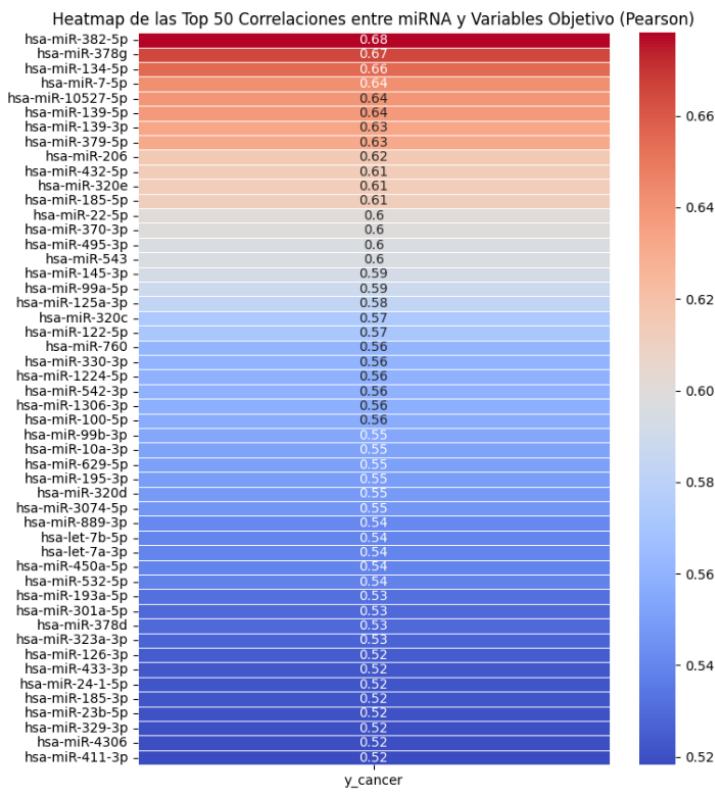


Figura 26: Top 50 correlaciones entre miARNs y Cáncer.

#### Gráfico de dispersión mediante PCA

El gráfico PCA mostró que las muestras de cáncer se agruparon de manera coherente y densa, formando patrones, lo que podría proporcionar indicios de subtipos biológicos en el cáncer, como diferentes tipos de cáncer, recolección de fluidos, etc. (Figura 27). Sin embargo, en el dataset *metadata* de la base de datos *liqDB* no se dispone de una columna con el tipo de cáncer con el que se pueda hacer una clasificación no binaria.

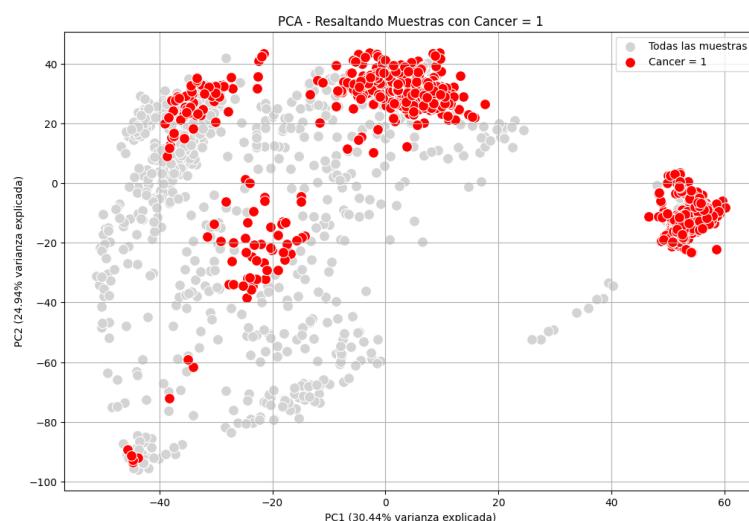


Figura 27: Cáncer - Gráfico de dispersión.

## 6.5. Evaluación de los miARNs seleccionados

Tras aplicar la selección de características, se obtuvieron 145 miARNs. El número de miARNs seleccionados por cada método se detalla en la Tabla 2. Los miARNs seleccionados se detallan en el archivo final\_selected\_features.csv, que se encuentra dentro del repositorio de GitHub. La ruta completa es: data/processed/final\_selected\_features.csv.

**Tabla 2:** Número de miARNs seleccionados por método.

Método	Número de características seleccionadas
Varianza baja	Todas
ANOVA	375
Información Mutua	300
RFECV + SVM	145
SVM + L1	179

Los miARNs seleccionados se evaluaron con los métodos listados a continuación:

### Comparación de las métricas de rendimiento

- **Área bajo la curva ROC (AUC-ROC):** La AUC-ROC después de la selección de características fue de 0,97, lo que refleja una capacidad casi perfecta del modelo para discriminar entre las clases. Antes de la selección, la AUC-ROC era de 0,96, lo que indica una ligera mejora en la capacidad de clasificación tras eliminar características irrelevantes o redundantes.
- **F1-Score:** El F1-Score se mantuvo constante antes y después de la selección de características, con un valor de 0,8676. Esto sugiere que el proceso de selección no afectó negativamente el balance entre estas dos métricas.
- **Precisión y Sensibilidad:** La precisión se mantuvo constante en 0,7917. La sensibilidad fue alta tanto antes como después de la selección, con un valor de 0,9596, lo que es crucial en un problema médico como la detección de cáncer.

### Curva de aprendizaje

El análisis de la curva de aprendizaje mostró indicios de sobreajuste (Figura 28). La curva de rendimiento en el conjunto de entrenamiento se mantuvo en 1,0, mientras que la curva de validación mejoró progresivamente hasta alcanzar aproximadamente 0,985, lo que sugiere que el modelo seguía aprendiendo de manera efectiva a medida que se añaden más datos, aunque existe una brecha que indica sobreajuste. Esto indica que, a pesar de que la selección de características ha reducido la complejidad del modelo, no ha sido suficiente para eliminar completamente el problema de sobreajuste. Este

problema se tratará en el apartado Construcción de modelos de clasificación con técnicas para mitigar el sobreajuste, como la regularización.

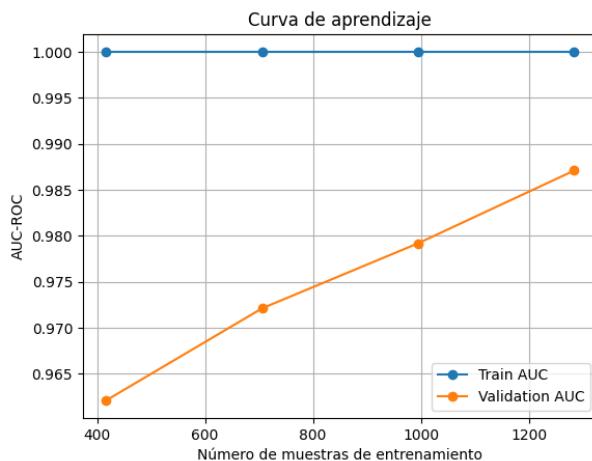


Figura 28: Curva de aprendizaje para evaluar el rendimiento de la selección de miARNs.

#### Análisis de correlación entre los miARNs seleccionadas

Aunque se redujo de manera considerable el número de correlaciones significativas, aún se observó una correlación alta entre varias de las características seleccionadas, lo que sugiere que podría ser posible reducir aún más el número de miARNs sin afectar gravemente el rendimiento del modelo (Figura 29).

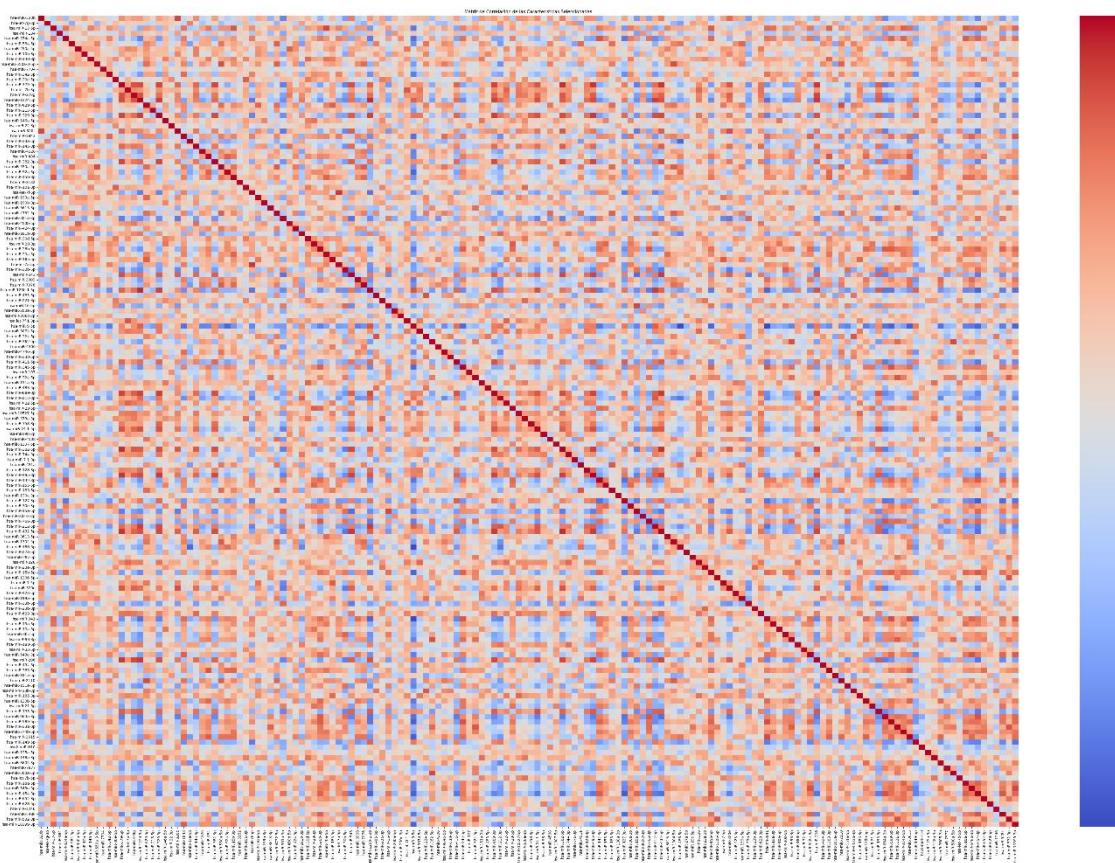


Figura 29: Mapa de calor con todas las correlaciones entre miARNs tras la selección de características.

#### Comparación del rendimiento antes y después de la selección de características

El rendimiento general del modelo, medido en términos de exactitud (*accuracy*), fue prácticamente idéntico antes y después de la selección de características, con valores de 0,94457 y 0,94458 respectivamente.

El proceso de selección de características fue eficaz para reducir la dimensionalidad del *dataset* sin comprometer el rendimiento del modelo. La ligera mejora en la AUC-ROC y el mantenimiento de otras métricas clave como el F1-Score sugieren que la selección de características ayudó a reducir el ruido en los datos y a mejorar la eficiencia del modelo. No obstante, el análisis de las curvas de aprendizaje indica que el modelo aún presenta un grado de sobreajuste, lo que justifica la necesidad de realizar ajustes adicionales a la hora de crear los modelos de clasificación.

En cuanto al número de características seleccionado, una regla empírica común es tener al menos 10 observaciones por cada característica. En este caso, en el *dataset* de entrenamiento tiene 1606 observaciones. El 10% equivale a 160,6, siendo un número mayor que las 154 características que tiene el *dataset* después de la selección.

La evaluación de los miARNs seleccionados ha sido muy completa y ha mostrado que la selección de características no solo redujo la complejidad del modelo, sino que mantuvo o mejoró ligeramente el rendimiento. A pesar de los resultados prometedores,

el análisis de la curva de aprendizaje y las correlaciones restantes indican que aún hay espacio para optimizaciones adicionales, especialmente en la reducción del sobreajuste y la redundancia de características.

## 6.6. Optimización de los modelos de clasificación

Se entrenaron y evaluaron los cinco modelos de clasificación. A continuación, se presentan los resultados y conclusiones para cada uno. Los detalles adicionales y los gráficos utilizados para el análisis se encuentran en el apartado Apéndice III: Curvas de aprendizaje. En este apartado, se han resumido los hiperparámetros seleccionados por cada modelo en la Tabla 3.

### SVM

El modelo SVM por defecto mostró un buen equilibrio entre entrenamiento y validación, con una mejora gradual en el rendimiento. El modelo optimizado con GridSearch presentó sobreajuste severo. La versión personalizada, al reducir el parámetro de regularización C, mejoró la estabilidad y alcanzó una puntuación de validación de 0,96, sin señales importantes de sobreajuste.

### KNN

El modelo KNN por defecto mostró un rendimiento sólido, con una buena capacidad de generalización. Sin embargo, la versión optimizada con GridSearch mostró un claro sobreajuste. El modelo personalizado mantuvo un rendimiento similar al original, mejorando la estabilidad de las predicciones y alcanzando una puntuación de validación de 0,94.

### XGBoost

El modelo XGBoost sin regularización presentó un sobreajuste significativo, mientras que la inclusión de regularización L1 y L2 mejoró ligeramente la estabilidad sin eliminar el problema por completo. El modelo optimizado con RandomSearch también presentó sobreajuste. El modelo personalizado 2, con una mayor regularización y ajuste de hiperparámetros, fue el más equilibrado y alcanzó una puntuación de validación de 0,95, aunque persisten indicios de sobreajuste.

### Random forest

Tanto el modelo *random forest* por defecto como el optimizado mostraron sobreajuste, con puntuaciones de entrenamiento perfectas. Sin embargo, el modelo personalizado, con ajustes en los hiperparámetros para reducir la complejidad del modelo, redujo notablemente el sobreajuste, mostrando una mejor capacidad de generalización con estabilidad en la puntuación de validación en torno a 0,94.

### MLP

El modelo MLP por defecto presentó inestabilidad y sobreajuste inicial. Tras la optimización de hiperparámetros con GridSearch, se logró una mejora significativa en la estabilidad del modelo y una convergencia en las puntuaciones de entrenamiento y validación en torno a 0,95, reduciendo la variabilidad.

*Resumen***Tabla 3:** Mejores hiperparámetros por modelo.

Modelo	Hiperparámetros	Métrica F1-Score
<b>SVM</b>	C = 10 kernel = 'rbf' degree = 2 gamma = 0,001	0,96
<b>KNN</b>	Algorithm = 'auto' n_neighbors = 3 weights = 'uniform'	0,94
<b>RF</b>	max_depth = 15 min_samples_leaf = 5 min_samples_split = 6 n_estimators=300	0,94
<b>XGB</b>	subsample = 0,8 reg_lambda = 5 reg_alpha=2 n_estimators = 300 max_depth = 5 learning_rate = 0,1 gamma = 0,1 colsample_bytree = 0,8	0,95
<b>MLP</b>	hidden_layer_sizes = (150, 100) activation = 'relu' solver = 'sgd' alpha = 0.01 batch_size = 32 learning_rate = 'adaptive' learning_rate_init = 0.01 max_iter = 200 early_stopping = True	0,95

## 6.7. Elección del mejor modelo de clasificación

A continuación, se presentan los resultados de los modelos de clasificación evaluados de manera resumida, destacando aquellos con mejor desempeño. Los detalles técnicos adicionales y las figuras complementarias se encuentran en el apartado Apéndice IV: Evaluación de modelos. En este apartado, se han resumido los resultados en la Tabla 4.

### *SVM*

El modelo SVM alcanzó una sensibilidad de 0,9596 y una precisión de 0,8261, con un f1-score de 0,8879. Estos valores sugieren un buen equilibrio entre precisión y sensibilidad, aunque el número de falsos positivos fue algo superior en comparación con otros modelos. Su AUC-ROC de 0,9788 refleja una excelente capacidad discriminativa,

mientras que la curva *Precision-Recall* mostró una alta consistencia en los valores de precisión conforme aumentaba la sensibilidad.

#### *KNN*

El modelo KNN mostró una sensibilidad de 0,9495 y una precisión de 0,7966, con un f1-score de 0,8664. Aunque su capacidad para detectar positivos fue adecuada, la precisión fue la más baja entre todos los modelos evaluados, lo que implicó un mayor número de falsos positivos. Su AUC-ROC fue de 0,9572, pero la curva *Precision-Recall* reflejó una caída considerable en la precisión conforme aumentaba la sensibilidad.

#### *Random forest*

El modelo *random forest* alcanzó una sensibilidad de 0,9596 y una precisión de 0,8190, con un f1-score de 0,8837. Aunque el modelo presentó un buen rendimiento global, fue superado en precisión por otros modelos. Su AUC-ROC fue de 0,9690, lo que indica un buen equilibrio entre clases, aunque la curva *Precision-Recall* mostró una menor capacidad para mantener una alta precisión con valores elevados de sensibilidad.

#### *XGBoost*

XGBoost fue el mejor modelo, con una sensibilidad de 0,9899, la más alta de todos, y una precisión de 0,8522, lo que resultó en un f1-score de 0,9159. Su AUC-ROC fue de 0,9775, reflejando una excelente capacidad discriminativa, y la curva *Precision-Recall* mostró una alta precisión, incluso con valores elevados de sensibilidad. Este modelo minimizó tanto los falsos negativos como los falsos positivos, convirtiéndose en el más adecuado para el contexto clínico.

#### *MLP*

El MLP presentó una sensibilidad de 0,9596 y una precisión de 0,8407, con un f1-score de 0,8962. Su rendimiento fue comparable al de SVM, aunque con una ligera ventaja en términos de sensibilidad. Su AUC-ROC fue de 0,9777, mientras que la curva *Precision-Recall* mostró una disminución notable en la precisión a medida que la sensibilidad aumentaba, situándolo por detrás de XGBoost en términos de fiabilidad general.

#### *Resumen*

En conjunto, el análisis muestra que **XGBoost** fue el mejor modelo de clasificación debido a su alta sensibilidad y precisión, además de su capacidad para minimizar tanto falsos negativos como falsos positivos. La elección de este modelo se justifica en el contexto clínico, donde maximizar la sensibilidad sin comprometer la precisión es crucial para la detección temprana del cáncer.

**Tabla 4:** Métricas de evaluación por modelo.

Modelo	Precisión	Sensibilidad	F1-Score	AUC-ROC	AUC-PR
<b>SVM</b>	0,8261	0,9596	0,8879	0,9788	0,9498
<b>KNN</b>	0,7966	0,9495	0,8664	0,9572	0,8527

<b>RF</b>	0,8190	0,9596	0,8837	0,9690	0,9204
<b>XGB</b>	0,8522	0,9899	0,9159	0,9775	0,9444
<b>MLP</b>	0,8407	0,9596	0,8962	0,9777	0,9410

## 6.8. Importancia de miARNs

### *Random forest*

En el modelo random forest, los cinco miARNs más importantes fueron:

1. hsa-miR-382-5p con una importancia de 0,0584.
2. hsa-let-7b-5p con una importancia de 0,0544.
3. hsa-miR-185-5p con una importancia de 0,0490.
4. hsa-miR-378g con una importancia de 0,0397.
5. hsa-miR-4306 con una importancia de 0,0395.

Estos miARNs destacaron por su capacidad para reducir la impureza en los árboles de decisión, lo que indica su relevancia en el proceso de clasificación de las muestras.

### *XGBoost*

En el modelo XGBoost, los miARNs más relevantes fueron:

1. hsa-miR-185-5p con una importancia de 0,1809.
2. hsa-miR-378g con una importancia de 0,1164.
3. hsa-let-7b-5p con una importancia de 0,0830.
4. hsa-miR-148a-3p con una importancia de 0,0598.
5. hsa-miR-629-5p con una importancia de 0,0408.

En este caso, hsa-miR-185-5p resultó ser el miARN más relevante, mostrando una diferencia importante en comparación con los otros miARNs, por lo que tiene una mayor frecuencia de uso en las divisiones del árbol. También destaca el miARNs hsa-miR-378g.

Ambos enfoques han identificado los miARNs hsa-miR-185-5p, hsa-miR-378g y hsa-let-7b-5p como los más influyentes en la detección de cáncer a partir de biopsias líquidas. Si bien los dos modelos coincidieron en tres características, XGBoost mostró una mayor concentración de importancia en un conjunto reducido de miARNs, mientras que *random forest* distribuyó la importancia de manera más uniforme.

Por otro lado, si estos resultados se comparan con los obtenidos en la Figura 26: Top 50 correlaciones entre miARNs y Cáncer., se puede ver cómo los 3 miARNs más influyentes forman parte de este Top 50, hsa-miR-185-5p, hsa-miR-378g en correlación positiva y hsa-let-7b-5p en correlación negativa; por lo que los resultados obtenidos durante el estudio son coherentes los unos con los otros.



En resumen, el análisis de la importancia de los miARNs resalta un conjunto coherente de biomarcadores clave, validados a través de múltiples enfoques (modelos de clasificación y análisis de correlación), lo que refuerza su relevancia en el proceso de clasificación.

Las tablas con la importancia por cada miARN se encuentran dentro del repositorio de GitHub. La ruta completa es: [results/characteristics](#).

## 7. Conclusiones y Trabajo futuro

Los resultados obtenidos en este estudio muestran que el uso de modelos de clasificación aplicados a miARNs permite detectar el cáncer con una alta precisión y sensibilidad.

La selección de características ha sido fundamental para reducir la dimensionalidad del conjunto de datos, mejorando la capacidad del modelo para generalizar y disminuir el sobreajuste. Entre los cinco modelos evaluados, XGBoost ha mostrado el mejor rendimiento, destacándose en términos de sensibilidad (98,99%) y f1-score (0,9159), lo que lo convierte en el modelo más adecuado para este tipo de problema clínico. Además, la importancia de ciertos miARNs, como el hsa-miR-185-5p, hsa-miR-378g y hsa-let-7b-5p ha quedado clara en todos los enfoques, confirmando su relevancia en la clasificación de muestras de biopsias líquidas.

Sin embargo, aunque el proceso de selección de características ha sido eficaz, se ha detectado una alta correlación entre algunos miARNs seleccionados. Esto sugiere que aún es posible refinar el conjunto de características sin comprometer el rendimiento del modelo. De hecho, el análisis de la curva de aprendizaje reveló señales persistentes de sobreajuste, lo que sugiere la necesidad de explorar nuevas estrategias para reducir aún más este efecto.

En cuanto a las limitaciones, uno de los principales retos ha sido la falta de información sobre el tipo de cáncer. Esta restricción ha impedido realizar una clasificación más detallada y detectar tipos específicos de cáncer.

Para mejorar los resultados obtenidos y abordar las limitaciones encontradas, se sugieren las siguientes líneas de investigación futura:

- 1. Optimización adicional de los modelos:** Algunos modelos aún muestran señales de sobreajuste. Para abordar este problema, sería interesante explorar técnicas de regularización más avanzadas, como ElasticNet. Esta técnica combina regularización L1 y L2, lo que puede ayudar a mejorar la generalización del modelo. Además, se podría considerar la implementación de métodos en cascada para la selección de características. Estos métodos permiten una selección más refinada al aplicar múltiples etapas de filtrado, lo que puede reducir el riesgo de incluir características irrelevantes o redundantes y reducir aún más el número de características seleccionadas. Finalmente, se podría evaluar el uso de redes neuronales profundas, ya que en la literatura se ha demostrado que pueden ofrecer un enfoque potente para capturar patrones complejos en los datos (Alharbi & Vakanski, 2023).
- 2. Verificación exhaustiva de las técnicas de preprocesamiento:** Realizar una comparación entre los modelos construidos con las técnicas de preprocesamiento, como la normalización, el tratamiento de valores faltantes y la gestión de valores atípicos, y aquellos construidos con los datos en crudo,

podría proporcionar información valiosa sobre la efectividad de estos métodos en el proyecto. Este análisis permitiría evaluar cómo cada técnica contribuye a mejorar el rendimiento de los modelos, facilitando la identificación de los enfoques más efectivos para el tratamiento de los datos.

3. **Ampliación del conjunto de datos:** El primer objetivo del trabajo fue desarrollar un algoritmo de clasificación multiclase para detectar diferentes tipos de cáncer. Sin embargo, el grupo de investigación dejó de actualizar la base de datos *liqDB* y este objetivo no se pudo llevar a cabo. Por lo tanto, uno de los aspectos clave para investigaciones futuras será la obtención de datos más ricos en cuanto a la clasificación de diferentes tipos de cáncer. Esto permitiría realizar un análisis más exhaustivo y mejorar la capacidad del modelo para discriminar entre diferentes tipos de cáncer, lo cual tiene una gran relevancia en la práctica clínica.
4. **Inspección de muestras provenientes del fluido de sangre:** En los gráficos de PCA realizados durante el análisis exploratorio de datos, se observó que las muestras provenientes del fluido de sangre son predominantes. Estas muestras se distribuyen principalmente a lo largo del primer componente principal, lo que indica que contienen la mayor parte de la información relevante para el análisis.
5. **Estudio longitudinal:** La inclusión de datos longitudinales permitiría evaluar la evolución temporal de los niveles de miARNs en pacientes, proporcionando así información adicional sobre la progresión de la enfermedad y la efectividad de los tratamientos. Esto podría derivar en un sistema de predicción más dinámico y útil en la práctica médica.
6. **Implementación clínica:** Aunque los resultados son prometedores, sería conveniente evaluar la viabilidad de implementar estos modelos en un entorno clínico real. Esto incluiría estudios de validación en cohortes más amplias y diversos centros médicos para garantizar su robustez y generalización en distintos contextos.

En conclusión, el presente estudio ha demostrado la viabilidad del uso de miARNs en la detección del cáncer a través de modelos de clasificación. Si bien se han logrado avances significativos, el trabajo futuro permitirá afinar aún más los modelos y acercarlos a una posible aplicación clínica.

## Referencias

- Aaltonen, K. E., Novosadová, V., Bendahl, P.-O., Graffman, C., Larsson, A.-M., Rydén, L., Aaltonen, K. E., Novosadová, V., Bendahl, P.-O., Graffman, C., Larsson, A.-M., & Rydén, L. (2017). Molecular characterization of circulating tumor cells from patients with metastatic breast cancer reflects evolutionary changes in gene expression under the pressure of systemic therapy. *Oncotarget*, 8(28), 45544-45565. <https://doi.org/10.18632/ONCOTARGET.17271>
- Abrams, Z. B., Johnson, T. S., Huang, K., Payne, P. R. O., & Coombes, K. (2019). A protocol to evaluate RNA sequencing normalization methods. *BMC Bioinformatics*, 20(24), 1-7. <https://doi.org/10.1186/S12859-019-3247-X/FIGURES/3>
- Alberts, B., Heald, R., Johnson, A., Morgan, D., Raff, M., Roberts, K., Walter, P., Wilson, J., & Hunt, T. (2022). *Molecular Biology of the Cell* (Seventh). W. W. Norton & Company.
- Alharbi, F., & Vakanski, A. (2023). Machine Learning Methods for Cancer Classification Using Gene Expression Data: A Review. *Bioengineering* 2023, Vol. 10, Page 173, 10(2), 173. <https://doi.org/10.3390/BIOENGINEERING10020173>
- American Cancer Society. (2024). *Imaging (Radiology) Tests for Cancer*. <https://www.cancer.org/cancer/diagnosis-staging/tests/imaging-tests/imaging-radiology-tests-for-cancer.html>
- Amezquita, R. A., Lun, A. T. L., Becht, E., Carey, V. J., Carpp, L. N., Geistlinger, L., Marini, F., Rue-Albrecht, K., Risso, D., Soneson, C., Waldron, L., Pagès, H., Smith, M. L., Huber, W., Morgan, M., Gottardo, R., & Hicks, S. C. (2019). Orchestrating single-cell analysis with Bioconductor. *Nature Methods* 2019 17:2, 17(2), 137-145. <https://doi.org/10.1038/s41592-019-0654-x>
- Anders, S., McCarthy, D. J., Chen, Y., Okoniewski, M., Smyth, G. K., Huber, W., & Robinson, M. D. (2013). Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. *Nature Protocols* 2013 8:9, 8(9), 1765-1786. <https://doi.org/10.1038/nprot.2013.099>
- Andreini, P., Bonechi, S., Bianchini, M., & Geraci, F. (2022). MicroRNA signature for interpretable breast cancer classification with subtype clue. *Journal of Computational Mathematics and Data Science*, 3, 100042. <https://doi.org/10.1016/J.JCMDS.2022.100042>
- Aparicio-Puerta, E., Jáspez, D., Lebrón, R., Koppers-Lalic, D., Marchal, J. A., & Hackenberg, M. (2019). liqDB: a small-RNAseq knowledge discovery database for liquid biopsy studies. *Nucleic Acids Research*, 47(D1), D113-D120. <https://doi.org/10.1093/NAR/GKY981>

- Bartel, D. P. (2004). MicroRNAs: Genomics, Biogenesis, Mechanism, and Function. *Cell*, 116(2), 281-297. [https://doi.org/10.1016/S0092-8674\(04\)00045-5](https://doi.org/10.1016/S0092-8674(04)00045-5)
- Bell, M. L., Fiero, M., Horton, N. J., & Hsu, C. H. (2014). Handling missing data in RCTs; A review of the top medical journals. *BMC Medical Research Methodology*, 14(1), 1-8. <https://doi.org/10.1186/1471-2288-14-118/TABLES/4>
- Bray Bsc, F., Laversanne, | Mathieu, Hyuna, |, Phd, S., Ferlay, J., Siegel Mph, R. L., Soerjomataram, I., Ahmedin, |, & Dvm, J. (2024). Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 74(3), 229-263. <https://doi.org/10.3322/CAAC.21834>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. <https://doi.org/10.1023/A:1010933404324/METRICS>
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. En *Classification and Regression Trees*. CRC Press. <https://doi.org/10.1201/9781315139470/CLASSIFICATION-REGRESSION-TREES-LEO-BREIMAN-JEROME-FRIEDMAN-OLSHEN-CHARLES-STONE/ACCESSIBILITY-INFORMATION>
- Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. *SIGMOD 2000 - Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93-104. <https://doi.org/10.1145/342009.335388>
- Bullard, J. H., Purdom, E., Hansen, K. D., & Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, 11(1), 1-13. <https://doi.org/10.1186/1471-2105-11-94/FIGURES/8>
- Calin, G. A., & Croce, C. M. (2006). MicroRNA signatures in human cancers. *Nature reviews. Cancer*, 6(11), 857-866. <https://doi.org/10.1038/NRC1997>
- Cancer Today | IARC. (2024, febrero 8). *Age-Standardized Rate (World) per 100 000, Mortality, Both sexes, in 2022*. <https://gco.iarc.who.int/today/en>
- Cancer Tomorrow | IARC. (2024, agosto 2). *Estimated number of new cases from 2022 to 2040, Both sexes, age [0-85+]*. <https://gco.iarc.fr/tomorrow/en>
- Cardinali, B., Tasso, R., Piccioli, P., Ciferri, M. C., Quarto, R., & Del Mastro, L. (2022). Circulating miRNAs in Breast Cancer Diagnosis and Prognosis. *Cancers*, 14(9). <https://doi.org/10.3390/CANCERS14092317>
- Casalino, G., Castellano, G., Consiglio, A., Nuzziello, N., & Vessio, G. (2023). MicroRNA expression classification for pediatric multiple sclerosis identification. *Journal of*

*Ambient Intelligence and Humanized Computing*, 14(12), 15851-15860.  
<https://doi.org/10.1007/S12652-021-03091-2/TABLES/6>

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321-357. <https://doi.org/10.1613/JAIR.953>

Chen, L., Heikkinen, L., Wang, C., Yang, Y., Sun, H., & Wong, G. (2019). Trends in the development of miRNA bioinformatics tools. *Briefings in Bioinformatics*, 20(5), 1836-1852. <https://doi.org/10.1093/BIB/BBY054>

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-August-2016, 785-794. <https://doi.org/10.1145/2939672.2939785>

Chen, X., Ba, Y., Ma, L., Cai, X., Yin, Y., Wang, K., Guo, J., Zhang, Y., Chen, J., Guo, X., Li, Q., Li, X., Wang, W., Zhang, Y., Wang, J., Jiang, X., Xiang, Y., Xu, C., Zheng, P., ... Zhang, C. Y. (2008). Characterization of microRNAs in serum: a novel class of biomarkers for diagnosis of cancer and other diseases. *Cell Research* 2008 18:10, 18(10), 997-1006. <https://doi.org/10.1038/cr.2008.282>

Chen, X., Zhang, B., Wang, T., Bonni, A., & Zhao, G. (2020). Robust principal component analysis for accurate outlier sample detection in RNA-Seq data. *BMC Bioinformatics*, 21(1), 1-20. <https://doi.org/10.1186/S12859-020-03608-0/FIGURES/4>

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szczesniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., & Mortazavi, A. (2016). A survey of best practices for RNA-seq data analysis. *Genome Biology* 2016 17:1, 17(1), 1-19. <https://doi.org/10.1186/S13059-016-0881-8>

Cortes, C., Vapnik, V., & Saitta, L. (1995). Support-vector networks. *Machine Learning* 1995 20:3, 20(3), 273-297. <https://doi.org/10.1007/BF00994018>

Cover, T. M., & Hart, P. E. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13(1), 21-27. <https://doi.org/10.1109/TIT.1967.1053964>

Croce, C. M. (2009). Causes and consequences of microRNA dysregulation in cancer. *Nature Reviews Genetics* 2009 10:10, 10(10), 704-714. <https://doi.org/10.1038/nrg2634>

Crowley, E., Di Nicolantonio, F., Loupakis, F., & Bardelli, A. (2013). Liquid biopsy: monitoring cancer-genetics in the blood. *Nature Reviews Clinical Oncology* 2013 10:8, 10(8), 472-484. <https://doi.org/10.1038/nrclinonc.2013.110>

- Das, C., Dubey, A., & Rasool, A. (2022). Outlier Detection Techniques: A Comparative Study. *Lecture Notes in Electrical Engineering*, 869, 551-566. [https://doi.org/10.1007/978-981-19-0019-8\\_42](https://doi.org/10.1007/978-981-19-0019-8_42)
- Diaz, L. A., & Bardelli, A. (2014). Liquid Biopsies: Genotyping Circulating Tumor DNA. *Cancer*, 120(3), 579-586. <https://doi.org/10.1200/JCO.2012.45.2011>
- Dillies, M.-A., Consortium, on behalf of T. F. S., Rau, A., Consortium, on behalf of T. F. S., Aubert, J., Consortium, on behalf of T. F. S., Hennequet-Antier, C., Consortium, on behalf of T. F. S., Jeanmougin, M., Consortium, on behalf of T. F. S., Servant, N., Consortium, on behalf of T. F. S., Keime, C., Consortium, on behalf of T. F. S., Marot, G., Consortium, on behalf of T. F. S., Castel, D., Consortium, on behalf of T. F. S., Estelle, J., ... Consortium, on behalf of T. F. S. (2013). A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Briefings in Bioinformatics*, 14(6), 671-683. <https://doi.org/10.1093/BIB/BBS046>
- Duda, R. O., & Hart, P. E. (1973). *Pattern Classification and Scene Analysis* (First). Wiley.
- Esquela-Kerscher, A., & Slack, F. J. (2006). Oncomirs — microRNAs with a role in cancer. *Nature Reviews Cancer* 2006 6:4, 6(4), 259-269. <https://doi.org/10.1038/nrc1840>
- Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Knowledge Discovery and Data Mining*.
- Evans, C., Hardin, J., & Stoebel, D. M. (2018). Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions. *Briefings in Bioinformatics*, 19(5), 776-792. <https://doi.org/10.1093/BIB/BBX008>
- Ewens, W. J., & Grant, G. (2005). *Statistical Methods in Bioinformatics*. <https://doi.org/10.1007/B137845>
- Fang, R., Pouyanfar, S., Yang, Y., Chen, S. C., & Iyengar, S. S. (2016). Computational Health Informatics in the Big Data Age. *ACM Computing Surveys (CSUR)*, 49(1). <https://doi.org/10.1145/2932707>
- Fix, E., & Hodges, J. L. (1989). Discriminatory Analysis - Nonparametric Discrimination: Consistency Properties. *International Statistical Review*, 57(3), 238. <https://doi.org/10.2307/1403797>
- Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367-378. [https://doi.org/10.1016/S0167-9473\(01\)00065-2](https://doi.org/10.1016/S0167-9473(01)00065-2)

- Géron, A. (2017). Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow (2019, O'reilly). En *Hands-On Machine Learning with R*. O'Reilly Media. <https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>
- Griffith, M., Walker, J. R., Spies, N. C., Ainscough, B. J., & Griffith, O. L. (2015). Informatics for RNA Sequencing: A Web Resource for Analysis on the Cloud. *PLOS Computational Biology*, 11(8), e1004393. <https://doi.org/10.1371/JOURNAL.PCBI.1004393>
- Hanahan, D., & Weinberg, R. A. (2011). Hallmarks of cancer: The next generation. *Cell*, 144(5), 646-674. <https://doi.org/10.1016/J.CELL.2011.02.013> ASSET/2067D218-2368-451A-B4DA-CF9EF4807B23/MAIN.ASSETS/GR1.JPG
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer. <https://doi.org/10.1007/978-0-387-84858-7>
- Hausser, J., & Zavolan, M. (2014). Identification and consequences of miRNA–target interactions — beyond repression of gene expression. *Nature Reviews Genetics* 2014 15:9, 15(9), 599-612. <https://doi.org/10.1038/nrg3765>
- ICBP - SURVMARK2. (s. f.). *Age-standardized 5-year net survival, both sexes, by age groups, colon cancer, 2010-2014*. Recuperado 22 de septiembre de 2024, de <https://gco.iarc.who.int/survival/survmark/index.html>
- James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An Introduction to Statistical Learning*. Springer International Publishing. <https://doi.org/10.1007/978-3-031-38747-0>
- Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4). <https://doi.org/10.1145/2382577.2382579>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (s. f.). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. <https://doi.org/10.5555/3294996.3295074>
- Kelleher, J. D., MacNamee, Brian., & D'Arcy, Aoife. (2015). *Fundamentals of machine learning for predictive data analytics: Algorithms, Worked Examples, and Case Studies*.
- Keup, C., Mach, P., Aktas, B., Tewes, M., Kolberg, H. C., Hauch, S., Sprenger-Haussels, M., Kimmig, R., & Kasimir-Bauer, S. (2018). RNA Profiles of Circulating Tumor Cells and Extracellular Vesicles for Therapy Stratification of Metastatic Breast Cancer Patients. *Clinical Chemistry*, 64(7), 1054-1062. <https://doi.org/10.1373/CLINCHEM.2017.283531>
- Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. *Computational*

and *Structural Biotechnology Journal*, 13, 8-17.  
<https://doi.org/10.1016/J.CSBJ.2014.11.005>

Leung, C. K., Madill, E. W. R., Souza, J., & Zhang, C. Y. (2022). Towards Trustworthy Artificial Intelligence in Healthcare. *Proceedings - 2022 IEEE 10th International Conference on Healthcare Informatics, ICHI 2022*, 626-632.  
<https://doi.org/10.1109/ICHI54592.2022.00127>

Li, Y., & Kowdley, K. V. (2012). MicroRNAs in Common Human Diseases. *Genomics, Proteomics & Bioinformatics*, 10(5), 246-253.  
<https://doi.org/10.1016/J.GPB.2012.07.005>

Lin, W. C., & Tsai, C. F. (2020). Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, 53(2), 1487-1509.  
<https://doi.org/10.1007/S10462-019-09709-4/METRICS>

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 413-422.  
<https://doi.org/10.1109/ICDM.2008.17>

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12), 1-21.  
<https://doi.org/10.1186/S13059-014-0550-8/FIGURES/9>

Luecken, M. D., & Theis, F. J. (2019). Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6), 8746.  
[https://doi.org/10.15252/MSB.20188746/SUPPL\\_FILE/MSB188746-SUP-0003-DATASETEV1.ZIP](https://doi.org/10.15252/MSB.20188746/SUPPL_FILE/MSB188746-SUP-0003-DATASETEV1.ZIP)

Mahalanobis, P. C. (1936). *On the generalized distance in statistics*.

Maher, C. A., Kumar-Sinha, C., Cao, X., Kalyana-Sundaram, S., Han, B., Jing, X., Sam, L., Barrette, T., Palanisamy, N., & Chinnaiyan, A. M. (2009). Transcriptome sequencing to detect gene fusions in cancer. *Nature 2009* 458:7234, 458(7234), 97-101. <https://doi.org/10.1038/nature07638>

Martins, I., Ribeiro, I. P., Jorge, J., Gonçalves, A. C., Sarmento-Ribeiro, A. B., Melo, J. B., & Carreira, I. M. (2021). Liquid Biopsies: Applications for Cancer Diagnosis and Monitoring. *Genes 2021*, Vol. 12, Page 349, 12(3), 349.  
<https://doi.org/10.3390/GENES12030349>

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115-133.  
<https://doi.org/10.1007/BF02478259/METRICS>

Mitchell, P. S., Parkin, R. K., Kroh, E. M., Fritz, B. R., Wyman, S. K., Pogosova-Agadjanyan, E. L., Peterson, A., Noteboom, J., O'Briant, K. C., Allen, A., Lin, D. W., Urban, N., Drescher, C. W., Knudsen, B. S., Stirewalt, D. L., Gentleman, R.,

- Vessella, R. L., Nelson, P. S., Martin, D. B., & Tewari, M. (2008). Circulating microRNAs as stable blood-based markers for cancer detection. *Proceedings of the National Academy of Sciences of the United States of America*, 105(30), 10513-10518.  
[https://doi.org/10.1073/PNAS.0804549105/SUPPL\\_FILE/0804549105SI.PDF](https://doi.org/10.1073/PNAS.0804549105/SUPPL_FILE/0804549105SI.PDF)
- Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., & Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods* 2008 5:7, 5(7), 621-628. <https://doi.org/10.1038/nmeth.1226>
- Pant, N., Rakshit, S., Paul, S., & Saha, I. (2019). Genome-wide analysis of multi-view data of miRNA-seq to identify miRNA biomarkers for stomach cancer. *Journal of Biomedical Informatics*, 97, 103254. <https://doi.org/10.1016/J.JBI.2019.103254>
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2017). CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems, 2018-December*, 6638-6648. <https://arxiv.org/abs/1706.09516v5>
- Rezaee, K., Jeon, G., Khosravi, M. R., Attar, H. H., & Sabzevari, A. (2022). Deep learning-based microarray cancer classification and ensemble gene selection approach. *IET Systems Biology*, 16(3-4), 120-131. <https://doi.org/10.1049/SYB2.12044>
- Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1), 139-140. <https://doi.org/10.1093/BIOINFORMATICS/BTP616>
- Robinson, M. D., & Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3), 1-9. <https://doi.org/10.1186/GB-2010-11-3-R25/FIGURES/3>
- Sarkar, J. P., Saha, I., Sarkar, A., & Maulik, U. (2021). Machine learning integrated ensemble of feature selection methods followed by survival analysis for predicting breast cancer subtype specific miRNA biomarkers. *Computers in Biology and Medicine*, 131, 104244. <https://doi.org/10.1016/J.COMPBIOMED.2021.104244>
- Schwarzenbach, H., Hoon, D. S. B., & Pantel, K. (2011). Cell-free nucleic acids as biomarkers in cancer patients. *Nature Reviews Cancer* 2011 11:6, 11(6), 426-437. <https://doi.org/10.1038/nrc3066>
- Sestini, S., Boeri, M., Marchiano, A., Pelosi, G., Galeone, C., Verri, C., Suatoni, P., Sverzellati, N., Vecchia, C. La, Sozzi, G., Pastorino, U., Sestini, S., Boeri, M., Marchiano, A., Pelosi, G., Galeone, C., Verri, C., Suatoni, P., Sverzellati, N., ... Pastorino, U. (2015). Circulating microRNA signature as liquid-biopsy to monitor

lung cancer in low-dose computed tomography screening. *Oncotarget*, 6(32), 32868-32877. <https://doi.org/10.18632/ONCOTARGET.5210>

Sherafatian, M. (2018). Tree-based machine learning algorithms identified minimal set of miRNA biomarkers for breast cancer diagnosis and molecular subtyping. *Gene*, 677, 111-118. <https://doi.org/10.1016/J.GENE.2018.07.057>

Srinivasu, P. N., Sandhya, N., Jhaveri, R. H., & Raut, R. (2022). From Blackbox to Explainable AI in Healthcare: Existing Tools and Case Studies. *Mobile Information Systems*, 2022(1), 8167821. <https://doi.org/10.1155/2022/8167821>

Stekhoven, D. J., & Bühlmann, P. (2012). MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1), 112-118. <https://doi.org/10.1093/BIOINFORMATICS/BTR597>

Stevens, W. L., Cox, D. R., & Scheffe, H. (1959). The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 21(1), 238-238. <https://doi.org/10.1111/J.2517-6161.1959.TB00334.X>

Tam, S., Tsao, M. S., & McPherson, J. D. (2015). Optimization of miRNA-seq data preprocessing. *Briefings in Bioinformatics*, 16(6), 950-963. <https://doi.org/10.1093/BIB/BBV019>

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., Van Baren, M. J., Salzberg, S. L., Wold, B. J., & Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology* 2010 28:5, 28(5), 511-515. <https://doi.org/10.1038/nbt.1621>

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., & Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6), 520-525. <https://doi.org/10.1093/BIOINFORMATICS/17.6.520>

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, 45(3), 1-67. <https://doi.org/10.18637/JSS.V045.I03>

Wagner, G. P., Kin, K., & Lynch, V. J. (2012). Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory in Biosciences*, 131(4), 281-285. <https://doi.org/10.1007/S12064-012-0162-3/METRICS>

Wan, J. C. M., Massie, C., Garcia-Corbacho, J., Mouliere, F., Brenton, J. D., Caldas, C., Pacey, S., Baird, R., & Rosenfeld, N. (2017). Liquid biopsies come of age: towards implementation of circulating tumour DNA. *Nature Reviews Cancer* 2017 17:4, 17(4), 223-238. <https://doi.org/10.1038/nrc.2017.7>

- Wang, C., Gao, X., & Liu, J. (2020). Impact of data preprocessing on cell-type clustering based on single-cell RNA-seq data. *BMC Bioinformatics*, 21(1), 1-13. <https://doi.org/10.1186/S12859-020-03797-8/FIGURES/4>
- Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics* 2008 10:1, 10(1), 57-63. <https://doi.org/10.1038/nrg2484>
- Watson, J. D., Baker, T. A., Bell, S. P., Gann, A. A. F., Levine, M., & Losick, R. M. (2013). *Molecular Biology of the Gene* (Seventh). Pearson.
- Weinberg, R. A. (2023). *The Biology of Cancer* (Third). W. W. Norton & Company.
- Weiss, S., Xu, Z. Z., Peddada, S., Amir, A., Bittinger, K., Gonzalez, A., Lozupone, C., Zaneveld, J. R., Vázquez-Baeza, Y., Birmingham, A., Hyde, E. R., & Knight, R. (2017). Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome*, 5(1), 1-18. <https://doi.org/10.1186/S40168-017-0237-Y/FIGURES/8>
- Yang, Y., Huang, N., Hao, L., & Kong, W. (2017). A clustering-based approach for efficient identification of microRNA combinatorial biomarkers. *BMC Genomics*, 18(2), 1-14. <https://doi.org/10.1186/S12864-017-3498-8/TABLES/8>
- Zhao, Y., Li, M. C., Konaté, M. M., Chen, L., Das, B., Karlovich, C., Williams, P. M., Evrard, Y. A., Doroshow, J. H., & McShane, L. M. (2021). TPM, FPKM, or Normalized Counts? A Comparative Study of Quantification Measures for the Analysis of RNA-seq Data from the NCI Patient-Derived Models Repository. *Journal of translational medicine*, 19(1). <https://doi.org/10.1186/S12967-021-02936-W>

# Apéndice I: Estadísticas descriptivas

*metadata antes del preprocessamiento*

**Tabla 5:** metadata - Estadísticas descriptivas antes del procesamiento.

	<b>SRP</b>	<b>Experiment</b>	<b>Sample</b>	<b>Instrument</b>
<b>count</b>	1606	1606	1606	1606
<b>unique</b>	30	1606	1603	9
<b>top</b>	SRP110505	SRX3848883	SAMN00009845	Illumina HiSeq 2000
<b>freq</b>	450	1	3	767

**Tabla 6:** metadata - Estadísticas descriptivas antes del procesamiento.

	<b>Sex</b>	<b>Fluid</b>	<b>Extraction</b>	<b>Library</b>
<b>count</b>	1130	1603	1518	1397
<b>unique</b>	2	21	7	3
<b>top</b>	female	plasma	miRNeasy	Illumina
<b>freq</b>	591	442	685	781

**Tabla 7:** metadata - Estadísticas descriptivas antes del procesamiento.

	<b>Healthy</b>	<b>Cancer</b>	<b>Exosome</b>	<b>Desc</b>
<b>count</b>	1603	1578	1606	1033
<b>unique</b>	2	2	2	104
<b>top</b>	True	False	False	Colorectal Cancer
<b>freq</b>	1002	1085	1238	100

## Apéndice II: Detección de valores atípicos

### *miARN*

#### *Isolation forest (IF)*

El propósito de IF es identificar puntos que están aislados en el espacio multidimensional. A medida que se incrementa la tasa de contaminación, el modelo tiende a etiquetar más puntos como *outliers*, lo cual es útil si se espera una mayor proporción de datos atípicos. Se realizaron gráficos PCA por cada configuración con el objetivo determinar cuál era la mejor para detectar valores atípicos en el *dataset* de *mirna* (Figura 30).

#### `n_estimators = 300, contamination = 0.01, 13 outliers`

Con una tasa de contaminación del 1%, el modelo ha identificado solo 13 *outliers*, marcados en rojo. Este nivel de contaminación parece muy bajo, ya que detecta muy pocos puntos como *outliers*. Aunque algunos de ellos están algo apartados de los otros puntos en el espacio de PCA, el número reducido podría significar que la detección es demasiado conservadora y tal vez esté perdiendo otros puntos que podrían ser *outliers* reales.

#### `n_estimators = 300, contamination = 0.05, 61 outliers`

Al incrementar la tasa de contaminación al 5%, se identifican 61 *outliers*. Aquí ya se empiezan a ver grupos más definidos de puntos rojos, pero algunos de ellos están mezclados con los puntos normales. Este nivel de contaminación podría ser más razonable si los datos realmente tuvieran una baja proporción de *outliers*, pero no garantiza una clara separación entre normales y anómalos.

#### `n_estimators = 300, contamination = 0.01, 122 outliers`

En este gráfico, con una tasa de contaminación del 10%, se identifica una mayor cantidad de puntos rojos. La cantidad de *outliers* parece ser excesiva, y algunos de ellos están mezclados con puntos que en el gráfico PCA parecen pertenecer a grupos más compactos de datos normales. Esto sugiere que el modelo empieza a marcar demasiados puntos como anómalos, posiblemente reduciendo la precisión.

#### `n_estimators = 300, contamination = 'auto', 206 outliers`

Al dejar que el parámetro de contaminación se ajuste automáticamente, se identifican muchos más *outliers*. En este caso, parece que el modelo está sobreestimando la cantidad. Muchos de ellos se encuentran distribuidos en grupos densos, lo que no es consistente con el comportamiento esperado de los *outliers*. Esta configuración parece la menos adecuada de las cuatro.

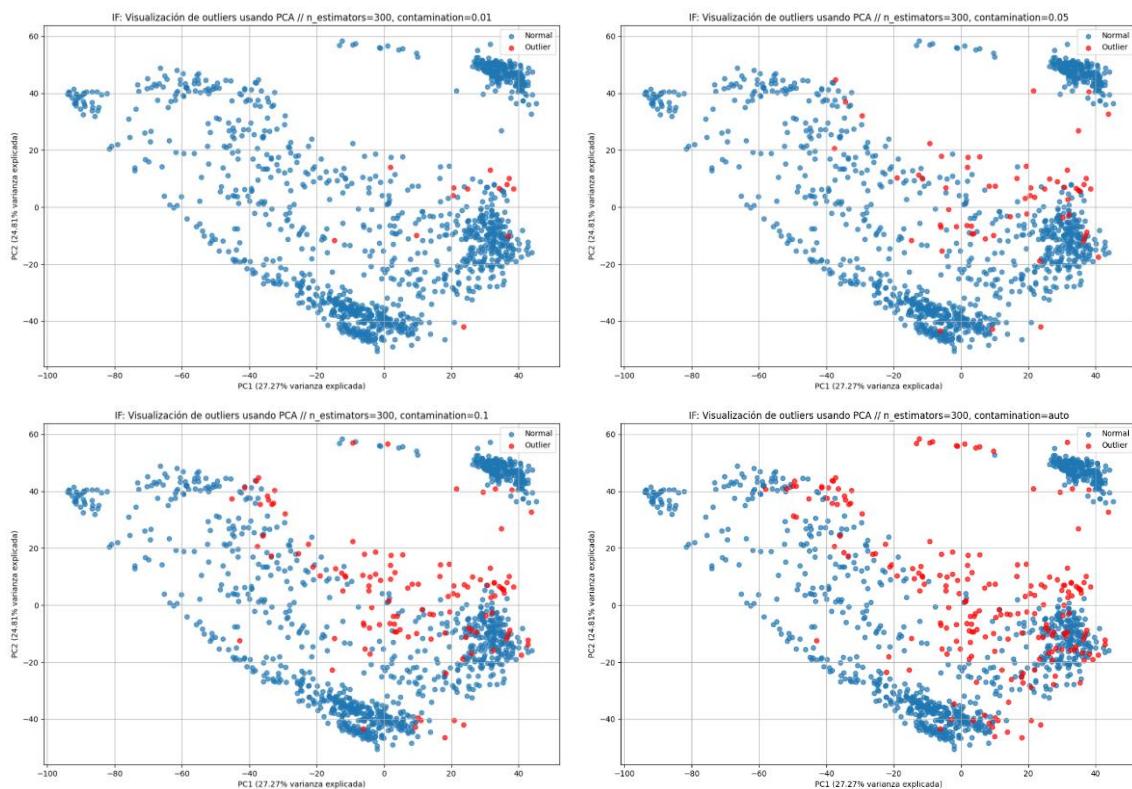


Figura 30: IF - Detección de valores atípicos.

#### *Local outlier factor (LOF)*

El número de vecinos en LOF afecta cómo se mide la densidad local. Con un valor bajo el método es más sensible a pequeñas diferencias en la densidad local, mientras que con valores más altos se tiene una perspectiva más amplia y menos sensible a pequeñas variaciones. En este caso también se realizaron gráficos con PCA para analizar las configuraciones (Figura 31).

#### *n\_neighbors = 15, contamination = 'auto', 49 outliers*

La configuración automática con 15 vecinos identificó 49 *outliers*. La mayoría de los *outliers* están dispersos a lo largo del gráfico, y algunos están algo alejados de los grupos principales, pero no parece haber una clara separación entre normales y *outliers* en todos los casos. Esta cantidad de *outliers* puede ser razonable, pero algunos parecen estar demasiado cerca de los puntos normales. Además, clasifica un grupo en la parte superior del gráfico como *outliers*.

#### *n\_neighbors = 20, contamination = 'auto', 47 outliers*

Aumentando el número de vecinos a 20, se identificaron 47 *outliers*, un número muy cercano al caso anterior. Al tener más vecinos, la densidad local se suaviza, lo que puede llevar a una detección más precisa en áreas con mayor densidad de puntos normales. Los resultados son similares al caso de 15 vecinos, pero algunos puntos fuera de los grupos principales parecen más justificados como *outliers*. Este también continúa detectando el grupo en la parte superior del gráfico.

***n\_neighbors = 10, contamination = 'auto', 41 outliers***

Con 10 vecinos, se detectaron 41 outliers. Menos vecinos implica que la detección de *outliers* es más local, lo que podría ser útil para detectar anomalías más pequeñas o específicas. La mayoría de los *outliers* están algo dispersos, con varios puntos ubicados lejos de los grupos principales, lo que sugiere que este nivel de contaminación podría estar ajustado adecuadamente. Sin embargo, algunos *outliers* han aparecido en áreas densas, lo que sugiere que tal vez este número de vecinos sea demasiado bajo para capturar la estructura global de los datos.

***n\_neighbors = 10, contamination = 0.01, 13 outliers***

Con 10 vecinos y una tasa de contaminación fija en 1%, el modelo detectó solo 13 *outliers*. La configuración es muy conservadora, lo que puede ser útil si solo se esperan pocos *outliers*. Sin embargo, algunos de los *outliers* detectados parecen estar demasiado cerca de los puntos normales, lo que podría indicar que el número de vecinos es demasiado pequeño para este nivel de contaminación.

***n\_neighbors = 10, contamination = 0.05, 61 outliers***

Con una tasa de contaminación del 5%, se detectaron 61 *outliers*. Los resultados son muy similares a la contaminación auto. Sin embargo, en este caso se han detectado más *outliers* en grupos densos.

***n\_neighbors = 10, contamination = 0.1, 122 outliers***

Con una tasa de contaminación del 10%, el número de *outliers* aumentó considerablemente. Al igual que con IF, con una tasa de contaminación alta, muchos puntos son etiquetados como *outliers*, incluidos algunos dentro de grupos densos de puntos normales. Esto sugiere que este nivel de contaminación puede ser demasiado elevado para este *dataset*.

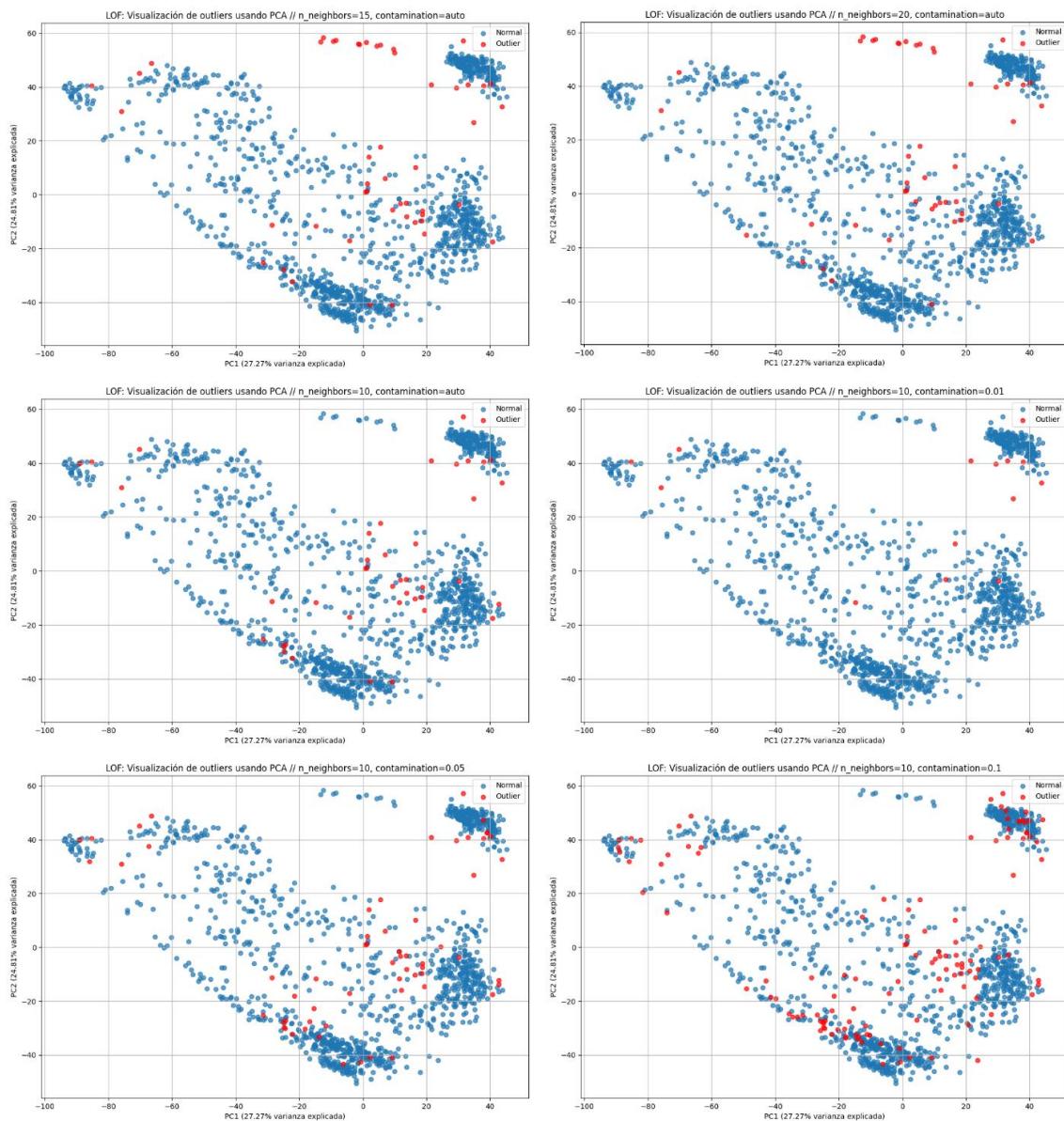


Figura 31: LOF - Detección de valores atípicos.

### DBSCAN

DBSCAN (*density-based spatial clustering of applications with noise*) es un algoritmo de agrupamiento basado en densidad que clasifica puntos como parte de un clúster si están rodeados por una cantidad suficiente de vecinos (*min\_samples*) dentro de una distancia específica (*eps*). Los puntos que no cumplen este criterio se consideran *outliers*.

En este caso, se determinaron los valores de *eps* = 30 y *min\_samples* = 10 mediante el método del codo.

Aunque DBSCAN es efectivo para detectar grupos de datos de forma arbitraria y *outliers* en conjuntos de datos con densidades bien diferenciadas, no fue efectivo en el caso del dataset de *mirna* porque los datos no presentaban diferencias de densidad (clústeres

compactos y regiones dispersas). Esto provocó que DBSCAN identificara un número excesivo de muestras como *outliers*, capturando falsos positivos en áreas menos densas que no representan desviaciones biológicas significativas (Figura 32).

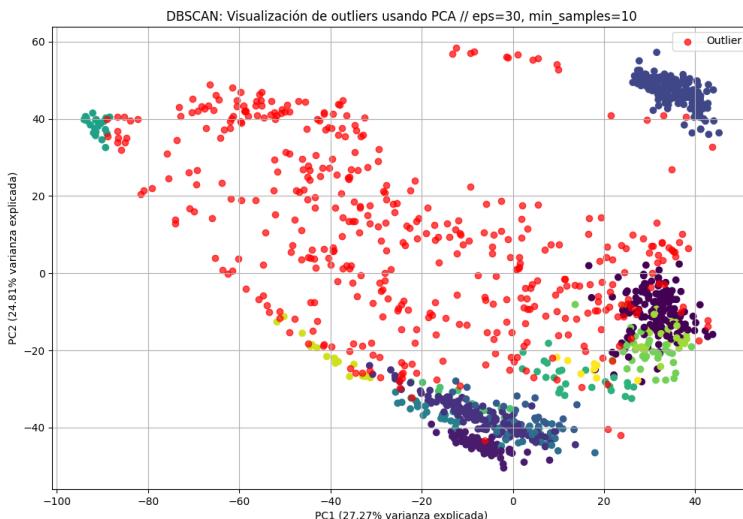


Figura 32: DBSCAN - Detección de valores atípicos.*Error! No se encuentra el origen de la referencia.*

#### Gráfico de Distancia Mahalanobis

Este gráfico muestra puntos etiquetados como normal y *outliers*, según la distancia Mahalanobis. El umbral del 97.5 percentil implica que se consideran *outliers* aquellos puntos cuya distancia esté en el 2.5% más alto. La mayoría de los *outliers* detectados están alejados del grupo denso de puntos normales, principalmente en las regiones más dispersas o separadas de la nube principal de datos (Figura 33).

La distancia de Mahalanobis es útil para identificar puntos que están inusualmente distantes del centro de la distribución, y este gráfico refleja eso de manera efectiva. Los *outliers* están bien distribuidos, mayormente en las áreas periféricas de la nube de puntos, lo que es coherente con lo que se esperaría de un análisis basado en esta métrica.

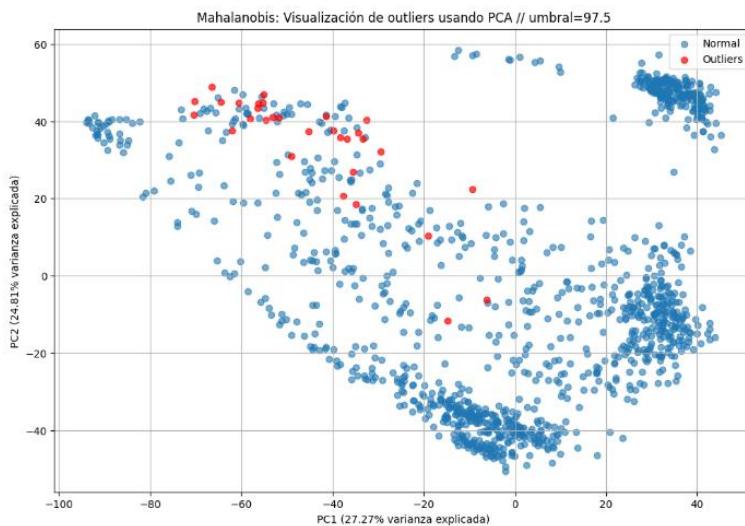


Figura 33: Mahalanobis - Detección de valores atípicos.

## *miARN y metadata*

En este caso, no se ha realizado un estudio comparativo con diferentes hiperparámetros.

### *Isolation Forest*

n\_estimators = 300, contamination = 'auto', 172 outliers

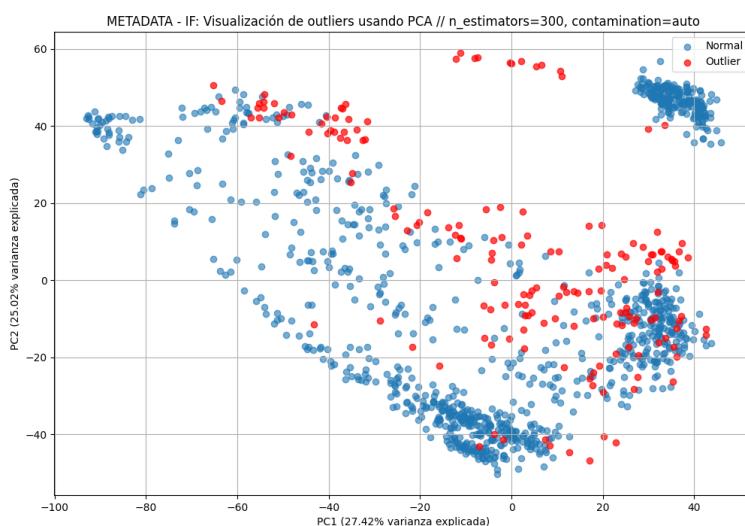


Figura 34: metadata - IF - Detección de valores atípicos.

*Local Outlier Factor*

n\_neighbors = 10, contamination = 'auto', 32 outliers

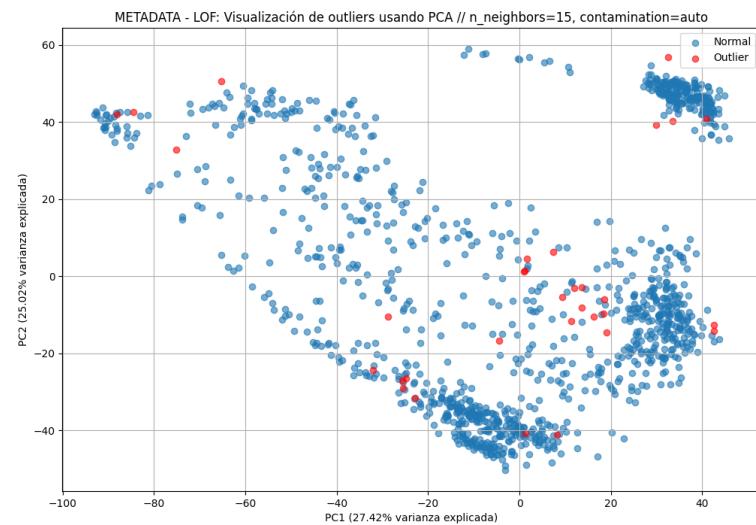


Figura 35: metadata - LOF - Detección de valores atípicos.

## Apéndice III: Curvas de aprendizaje

### SVM

#### SVM por defecto

El modelo por defecto de SVM muestra un buen balance entre entrenamiento y validación. La puntuación de entrenamiento comienza en ~0,87 y aumenta hasta ~0,94, mientras que la puntuación de validación crece de 0,88 a ~0,94 con más datos. La diferencia mínima entre ambas puntuaciones indica que el modelo no está sobreajustando y generaliza bien, sin signos de subajuste.

#### SVM con GridSearch

Hiperparámetros: C = 100, degree = 2, gamma = 0,001, kernel = 'rbf'

Al ajustar los hiperparámetros con GridSearch, el modelo presenta un claro sobreajuste. La puntuación de entrenamiento se mantiene en 1,0, lo que indica que el modelo está memorizando los datos, mientras que la puntuación de validación, aunque alto (~0,96), es inferior al de entrenamiento, lo que confirma el sobreajuste.

#### SVM personalizado

Hiperparámetros: C = 10, kernel = 'rbf', degree = 2, gamma = 0.001, probability = True

Con el objetivo de reducir el *overfitting*, se mantuvieron los hiperparámetros optimizados, pero se introdujo más regularización reduciendo el parámetro C. El modelo personalizado logra mejorar el desempeño general. La puntuación de entrenamiento comienza en ~0,97 y sube a ~0,98, mientras que el de validación mejora significativamente, llegando a ~0,96. Además, la reducción del área sombreada sugiere una mayor estabilidad, y no se observan grandes signos de sobreajuste.

### Conclusiones

El modelo por defecto ofrece un buen equilibrio entre entrenamiento y validación, mientras que el ajuste mediante GridSearch genera un fuerte sobreajuste. El modelo personalizado logra el mejor rendimiento general, mostrando una clara mejora en la estabilidad y en las puntuaciones de validación, sin sobreajustar los datos.

Los tres gráficos se pueden revisar en la Figura 36.

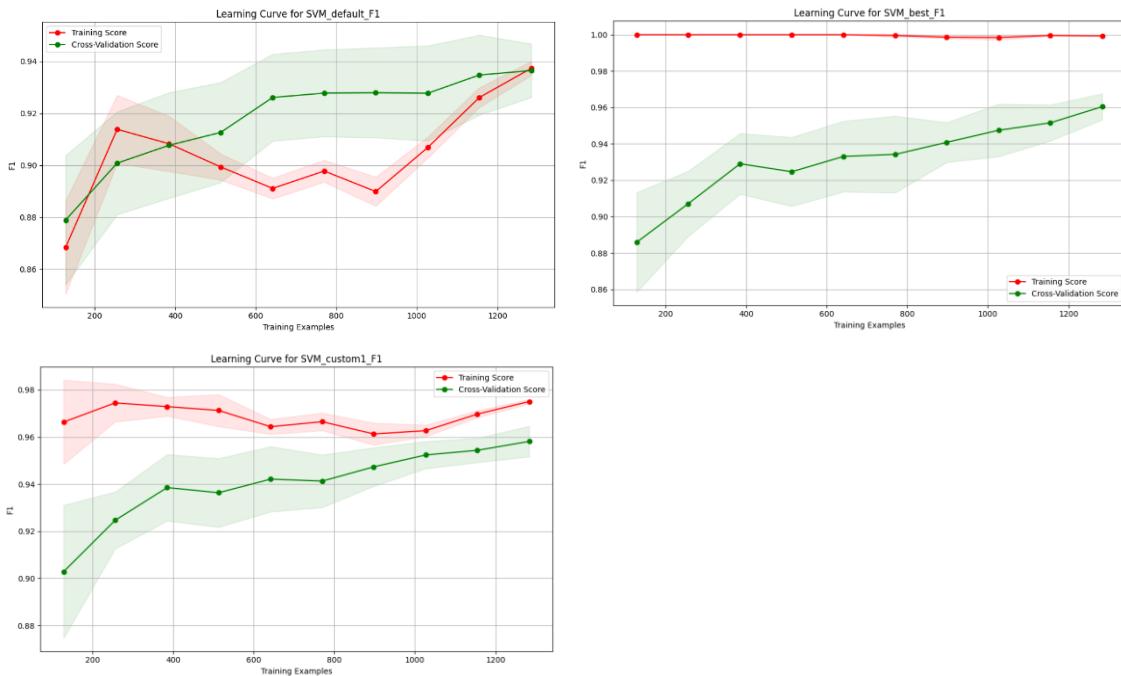


Figura 36: SVM - Curvas de aprendizaje - Modelo por defecto, modelo con optimización de hiperparámetros y modelo personalizado.

## KNN

### KNN por defecto

El modelo KNN por defecto tiene un buen rendimiento, con una puntuación de entrenamiento que comienza en ~0,87 y se estabiliza alrededor de 0,95. La puntuación de validación mejora de ~0,86 a ~0,94 conforme aumenta el número de ejemplos. La diferencia entre las puntuaciones de entrenamiento y validación es mínima, lo que indica una buena capacidad de generalización y ausencia de sobreajuste.

### KNN con GridSearch

Hiperparámetros: algorithm = 'auto', n\_neighbors = 3, weights = 'distance'

El modelo optimizado con GridSearch presenta un sobreajuste claro, con una puntuación de entrenamiento de 1,0; también aumenta la puntuación de validación (~0,95). Aunque la puntuación de validación es alta, la memorización de los datos de entrenamiento sugiere una pérdida de capacidad de generalización.

### KNN personalizado

Hiperparámetros: algorithm = 'auto', n\_neighbors = 3, weights = 'uniform'

El modelo personalizado mantiene el número de vecinos optimizado, pero usa pesos uniformes. Los resultados son similares al modelo por defecto, pero la reducción del área sombreada en la curva de validación refleja una mayor estabilidad, haciendo que el modelo sea más robusto.

## Conclusiones

El modelo por defecto ofrece un buen balance entre rendimiento y generalización, mientras que el modelo optimizado muestra un sobreajuste significativo. El modelo personalizado mejora la estabilidad al reducir la variabilidad en la curva de validación, lo que lo convierte en una opción más fiable frente a variaciones en los datos.

Los tres gráficos se pueden revisar en la Figura 37.

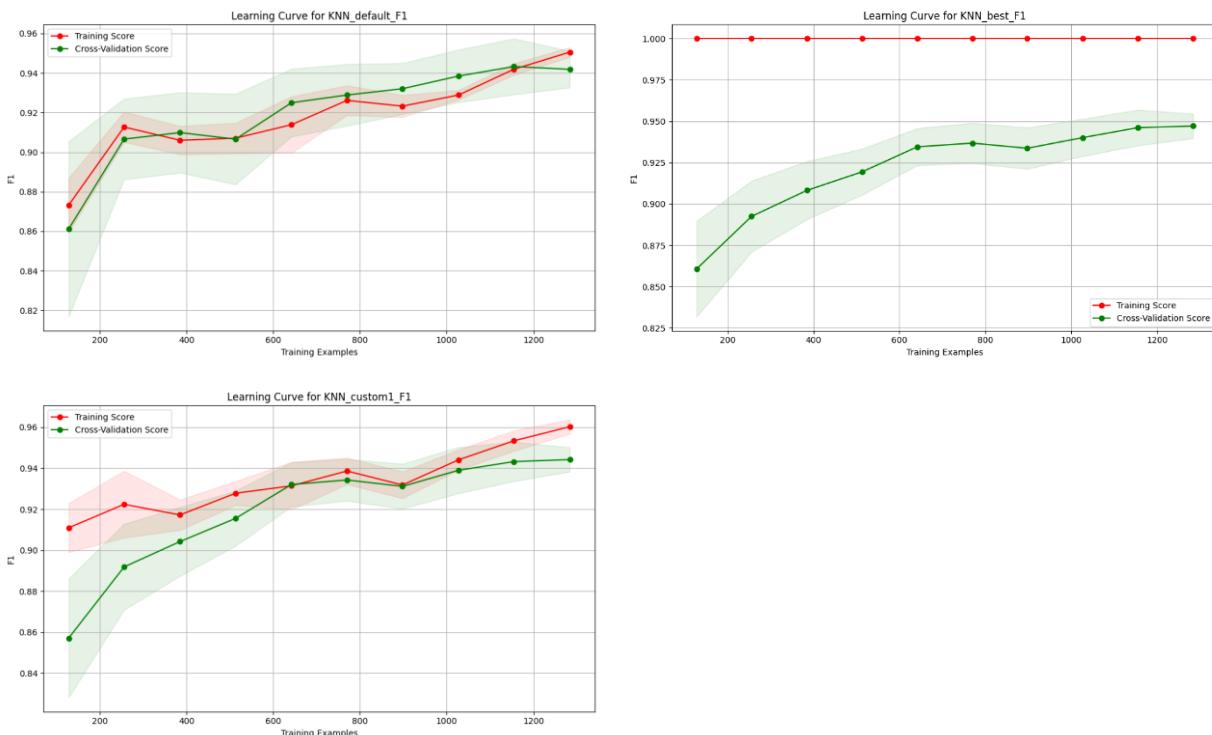


Figura 37: KNN - Curvas de aprendizaje - Modelo por defecto, modelo con optimización de hiperparámetros y modelo personalizado.

## XGBoost

Se ha optado por XGBoost, en vez de *Gradient Boosting* porque ofrece ventajas significativas como la velocidad y el rendimiento, ya que se puede ejecutar en GPU, y técnicas de regularización como L1 y L2, las cuales son muy importantes para tratar el sobreajuste de este tipo de modelos.

### XGBoost por defecto (sin regularización)

El modelo XGBoost sin regularización presenta un claro sobreajuste, con una puntuación de entrenamiento constantemente en 1,0. La puntuación de validación comienza en ~0,85 y mejora hasta ~0,95 a medida que aumenta el tamaño del conjunto de entrenamiento. Aunque el rendimiento en validación es aceptable, el sobreajuste significativo sugiere que el modelo está memorizando los datos de entrenamiento.

### XGBoost con regularización (L1 y L2)

Hiperparámetros: reg\_alpha = 0,1 (L2), reg\_lambda = 1,0 (L2)

El modelo regularizado sigue mostrando un alto score de entrenamiento (~1,0), pero se observa una pequeña disminución en algunos puntos, indicando que la regularización ha reducido ligeramente el sobreajuste. La puntuación de validación alcanza ~0,95, con una menor variabilidad que el modelo sin regularización, lo que refleja una mejora en la estabilidad.

#### XGBoost optimizado con Random Search

Hiperparámetros: subsample = 0,6, reg\_lambda = 0,1, reg\_alpha = 0,01, n\_estimators = 100, max\_depth = 7, learning\_rate = 0,3, gamma = 0, colsample\_bytree = 0,8

El modelo optimizado con Random Search sigue presentando sobreajuste, y el rendimiento no ha mejorado respecto al anterior. La puntuación de validación alcanza ~0,95, aunque la puntuación de entrenamiento permanece en 1,0, lo que indica que el problema de sobreajuste persiste.

#### XGBoost personalizado 1

Hiperparámetros: subsample = 0,6, reg\_lambda = 5, reg\_alpha = 2, n\_estimators = 100, max\_depth = 7, learning\_rate = 0,3, gamma = 0, colsample\_bytree = 0,8

En el modelo personalizado 1, se aumentó la regularización, manteniendo los demás hiperparámetros optimizados. Aunque el sobreajuste se redujo ligeramente, la curva de entrenamiento aún muestra indicios de sobreajuste a partir de 400 muestras, lo que sugiere que el modelo todavía memoriza algunos datos.

#### XGBoost personalizado 2

Hiperparámetros: subsample = 0,8, reg\_lambda = 5, reg\_alpha = 2, n\_estimators = 300, max\_depth = 5, learning\_rate = 0,1, gamma = 0,1, colsample\_bytree = 0,8

En el modelo personalizado 2, se realizaron ajustes significativos en los hiperparámetros con el objetivo de obtener un modelo menos complejo para reducir el *overfitting*. Este modelo presenta reducción en el sobreajuste, y la curva de validación alcanza ~0,95, lo que indica un rendimiento muy similar a los modelos anteriores. Aun así, sigue mostrando bastantes indicios de sobreajuste.

#### Conclusiones

Todos los modelos de XGBoost muestran algún nivel de sobreajuste. La regularización aplicada ayuda a reducir ligeramente este fenómeno, pero no lo elimina por completo. El mejor rendimiento general se alcanza con el modelo optimizado con personalizado 2. Para reducir aún más el sobreajuste, sería necesario aplicar una regularización más fuerte o explorar técnicas de reducción de dimensionalidad.

Los cuatro gráficos se pueden revisar en la Figura 38.

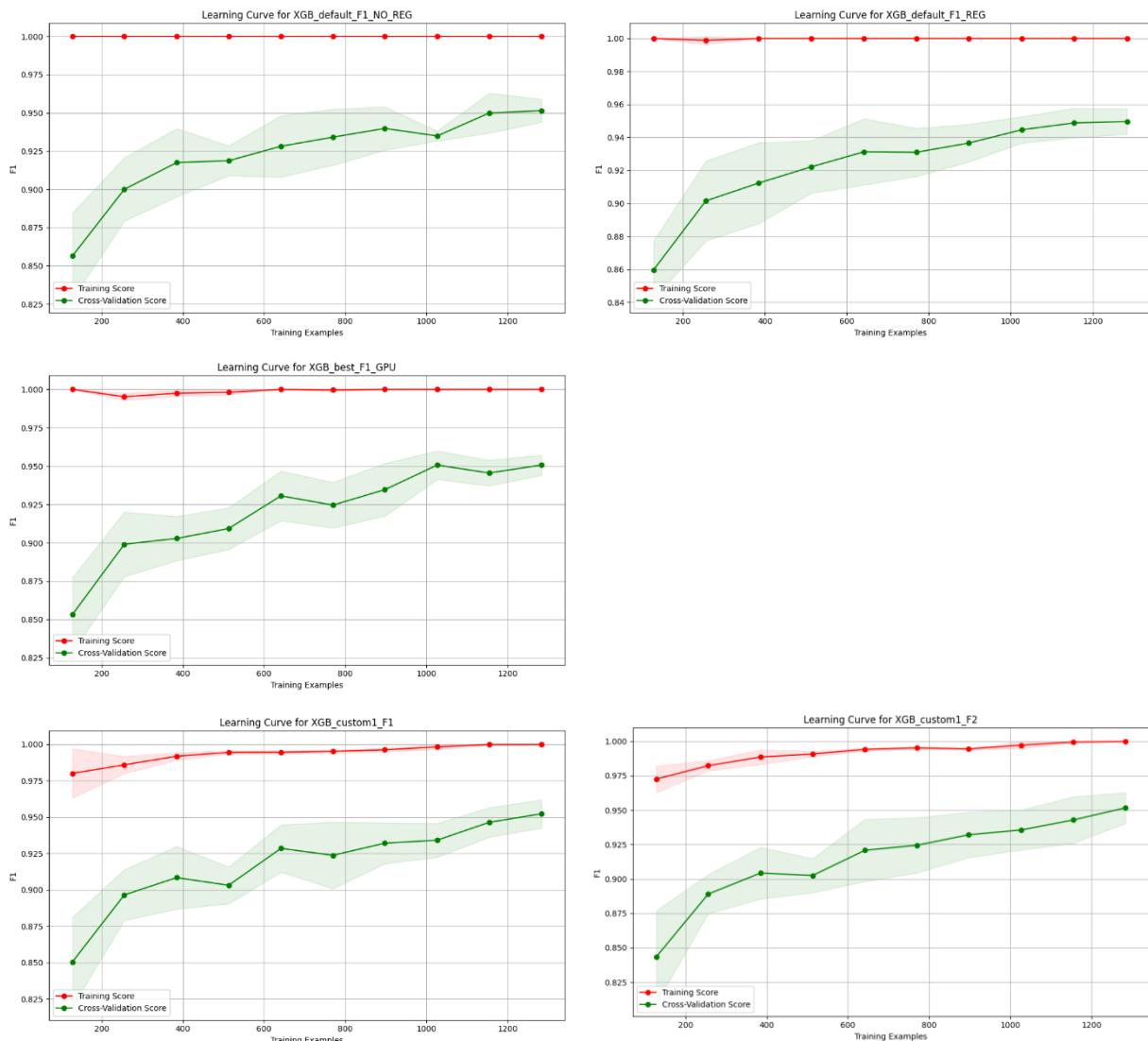


Figura 38: XGBoost - Curvas de aprendizaje - Modelo por defecto, Modelo por defecto regularizado, modelo con optimización de hiperparámetros, modelo personalizado 1 y modelo personalizado 2.

### Random forest

#### Ranfom forest por defecto

El modelo *random forest* por defecto muestra un claro sobreajuste, con una puntuación de entrenamiento que se mantiene en 1,0 en todos los tamaños de muestra. La puntuación de validación comienza alrededor de 0,85 y mejora, estabilizándose en aproximadamente 0,95. A pesar de la alta variabilidad inicial en el área sombreada de la curva de validación, esta disminuye a medida que se agregan más ejemplos, lo que indica una mejora en la estabilidad. Sin embargo, el modelo está memorizando los datos de entrenamiento.

#### Random Forest con GridSearch

Hiperparámetros: `max_depth = None`, `min_samples_leaf = 1`, `min_samples_split = 2`, `n_estimators = 500`

El modelo optimizado mediante GridSearch presenta un comportamiento similar al modelo por defecto, con una puntuación de entrenamiento que permanece en 1,0. La puntuación de validación también se estabiliza alrededor de 0,95, lo que refleja un rendimiento razonablemente bueno, aunque no se observan mejoras significativas en comparación con el modelo por defecto. La variabilidad en la validación sigue siendo alta al inicio, pero se reduce a medida que se incrementan los datos.

#### Random Forest personalizado

Hiperparámetros: max\_depth = 15, min\_samples\_leaf = 5, min\_samples\_split = 6, n\_estimators= 300

Para el modelo personalizado, se ajustaron manualmente los hiperparámetros con el objetivo de reducir el sobreajuste, modificando parámetros como la profundidad máxima de los árboles y el número mínimo de muestras para dividir nodos. Como resultado, el modelo muestra una considerable reducción en el sobreajuste, y ambas curvas (entrenamiento y validación) presentan una diferencia mínima al final, indicando una mejora en la capacidad de generalización. Además, la puntuación de validación se estabiliza alrededor de 0,94, obteniendo un buen resultado.

#### Conclusiones

Los modelos *Random Forest*, tanto el por defecto como el optimizado, exhiben un notable sobreajuste, con scores de entrenamiento consistentemente altos en comparación con la validación. A pesar de la optimización de hiperparámetros, el sobreajuste persiste. Sin embargo, el modelo personalizado, con ajustes manuales,

logra una mejor estabilidad y reduce significativamente el sobreajuste, mostrando una diferencia mínima entre las curvas de entrenamiento y validación.

Los tres gráficos se pueden revisar en la Figura 39.

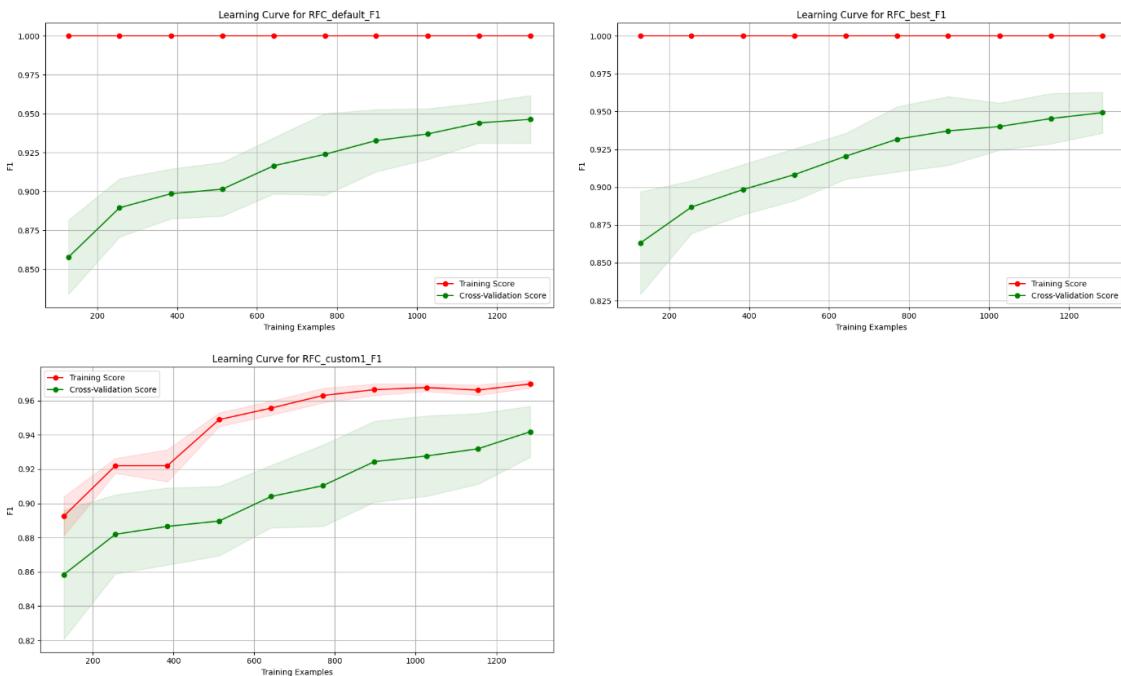


Figura 39: Random Forest - Curvas de aprendizaje - Modelo por defecto, modelo con optimización de hiperparámetros y modelo personalizado.

### Perceptrón multicapa

#### Perceptrón Multicapa por Defecto

El modelo MLP por defecto comienza con una puntuación de 0,99, lo que indica que el modelo está memorizando casi por completo los datos de entrenamiento. Se observa sobreajuste hasta que, tras aproximadamente 500 ejemplos, la puntuación disminuye, mostrando inestabilidad. Además, la variabilidad aumenta al disminuir la puntuación. La puntuación de validación comienza en 0,90 y mejora gradualmente hasta estabilizarse en torno a 0,95, aunque muestra cierta inestabilidad, especialmente con menos de 1000 ejemplos. Este modelo muestra *overfitting* y variabilidad.

#### Perceptrón Multicapa Optimizado con GridSearch

Hiperparámetros: `hidden_layer_sizes = (150, 100)`, `activation = 'relu'`, `solver = 'sgd'`, `alpha = 0,01`, `batch_size = 32`, `learning_rate = 'adaptive'`, `learning_rate_init = 0,01`, `max_iter = 200`, `early_stopping = True`

El modelo MLP optimizado mediante GridSearch muestra una mejora notable en términos de estabilidad y desempeño. La puntuación de entrenamiento comienza en un nivel más bajo (~0,85), lo que indica que el modelo no está sobreajustando desde el principio. A medida que se incrementa el número de ejemplos, la puntuación de entrenamiento aumenta y se estabiliza en ~0,95. La puntuación de validación sigue una

tendencia similar, comenzando en 0,85 y alcanzando 0,95, con una disminución significativa en la variabilidad observada en el modelo por defecto.

### Conclusiones

En el modelo por defecto, aunque no existe un sobreajuste severo, la inestabilidad observada en ambas curvas sugiere que el modelo podría beneficiarse de ajustes de hiperparámetros. El modelo optimizado con GridSearch, por otro lado, muestra una mejora clara, sin signos de sobreajuste y con puntuaciones de validación y entrenamiento que convergen en torno a 0,95. La reducción de la variabilidad en las puntuaciones de validación indica una mayor estabilidad y mejor generalización conforme se agregan más datos.

Los gráficos se pueden revisar en la Figura 40.

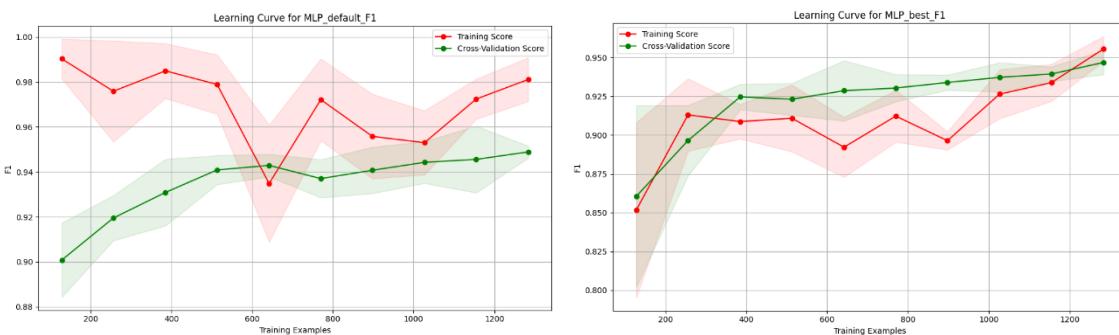


Figura 40: MLP - Curvas de aprendizaje - Modelo por defecto, modelo con optimización de hiperparámetros.

## Apéndice IV: Evaluación de modelos

### SVM

- **Precisión:** 0,8261
- **Sensibilidad:** 0,9596
- **F1-Score:** 0,8879
- **AUC-ROC:** 0,9788
- **AUC-PR:** 0,9498

El SVM muestra un muy buen desempeño general, con una alta sensibilidad (0,9596) lo que indica que es muy efectivo detectando los casos positivos (cáncer). Sin embargo, su precisión (0,8261) es moderada. La tasa de falsos negativos es baja (4), y su f1-score (0,8879) y AUC-ROC (0,9788) son bastante competitivos. Es un buen candidato por su equilibrio entre precisión y sensibilidad.

La curva ROC muestra una excelente capacidad discriminativa, con una tasa de verdaderos positivos alta y falsos positivos bajos. El AUC de 0,9788 indica que el modelo es muy eficaz en diferenciar entre clases.

La curva PR también es muy consistente. La precisión se mantiene alta incluso cuando la sensibilidad aumenta, lo que sugiere que este modelo maneja bien la clase positiva (cáncer).

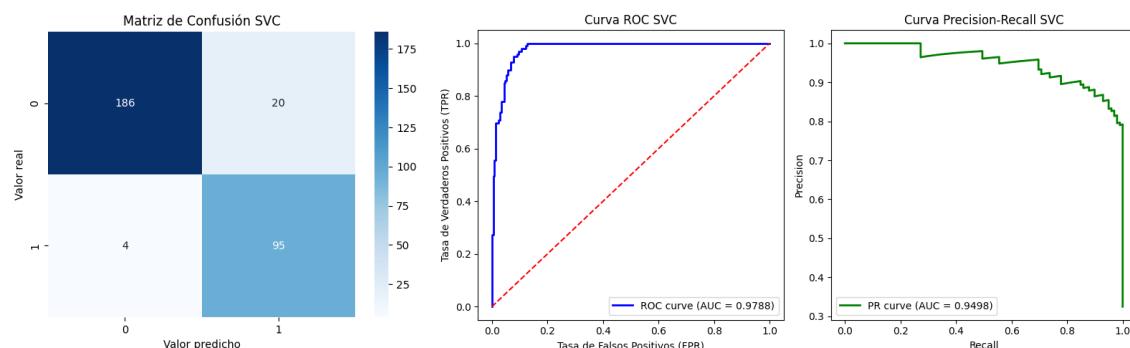


Figura 41: SVM - Matriz de confusión, curva ROC y curva PR.

### KNN

- **Precisión:** 0,7966
- **Sensibilidad:** 0,9495
- **F1-Score:** 0,8664
- **AUC-ROC:** 0,9572
- **AUC-PR:** 0,8527

KNN tiene una sensibilidad alto (0,9495), lo que sugiere que identifica la mayoría de los casos de cáncer, pero su precisión es la más baja entre todos los modelos (0,7966). Esto implica un mayor número de falsos positivos comparado con otros modelos.

Aunque su desempeño es aceptable, no es el mejor comparado con SVM y otros modelos en términos de precisión y f1-score.

Aunque la curva ROC indica un buen rendimiento con un AUC de 0,9572, no es tan robusta como la del SVC. Esto sugiere que este modelo puede ser algo menos efectivo en la clasificación de los pacientes.

La curva PR muestra una precisión decreciente conforme aumenta la sensibilidad, lo que indica que este modelo tiene más problemas en la clasificación correcta de los casos de cáncer en situaciones más difíciles.

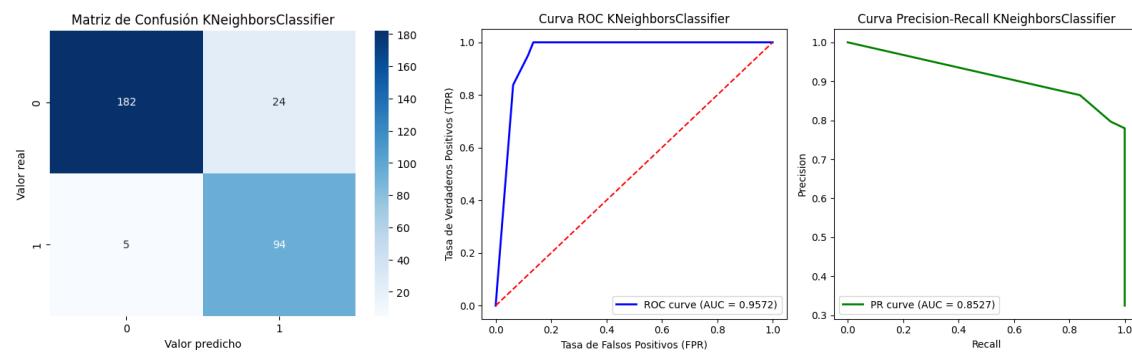


Figura 42: KNN - Matriz de confusión, curva ROC y curva PR.

#### Random Forest

- **Precisión:** 0,8190
- **Sensibilidad:** 0,9596
- **F1-Score:** 0,8837
- **AUC-ROC:** 0,9690
- **AUC-PR:** 0,9135

Random Forest tiene la misma sensibilidad que SVM (0,9596) pero una precisión ligeramente más baja (0,8190). Esto lo posiciona como un modelo fuerte en la detección de cáncer, aunque con un poco más de falsos positivos que SVM y XGBoost. Es confiable pero no destaca como el mejor en precisión.

La curva ROC es bastante fuerte con un AUC de 0,9690, lo que refleja un buen equilibrio entre la clasificación de pacientes sanos y de cáncer.

La curva PR muestra peores resultados, ya que la precisión no es tan alta como la del SVC o XGBoost en los valores de sensibilidad más elevados. Aun así, ofrece una buena capacidad para distinguir la clase minoritaria (cáncer).

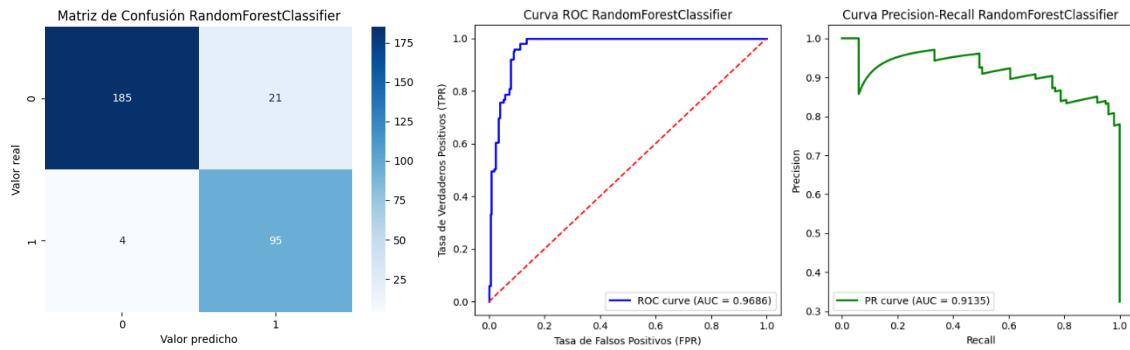


Figura 43: Random Forest: KNN - Matriz de confusión, curva ROC y curva PR.

#### XGBoost

- **Precisión:** 0,8522
- **Sensibilidad:** 0,9899
- **F1-Score:** 0,9159
- **AUC-ROC:** 0,9775
- **AUC-PR:** 0,9444

XGBoost sobresale en su capacidad de detectar casos de cáncer, con una sensibilidad de 0,9899, la más alto de todos los modelos. También tiene una precisión considerablemente alta (0,8522) y un f1-score sólido (0,9159), lo que lo convierte en el mejor candidato. El número de falsos negativos es extremadamente bajo, lo que es crucial en aplicaciones biomédicas.

La curva ROC muestra un excelente rendimiento con un AUC muy cercano al del SVC (0,9775), lo que sugiere que XGBoost es extremadamente eficaz en la clasificación.

La curva PR refleja una alta precisión incluso para valores altos de sensibilidad, lo que indica que XGBoost mantiene su capacidad predictiva a medida que intenta identificar más casos de cáncer, siendo muy confiable en la clasificación de la clase positiva.

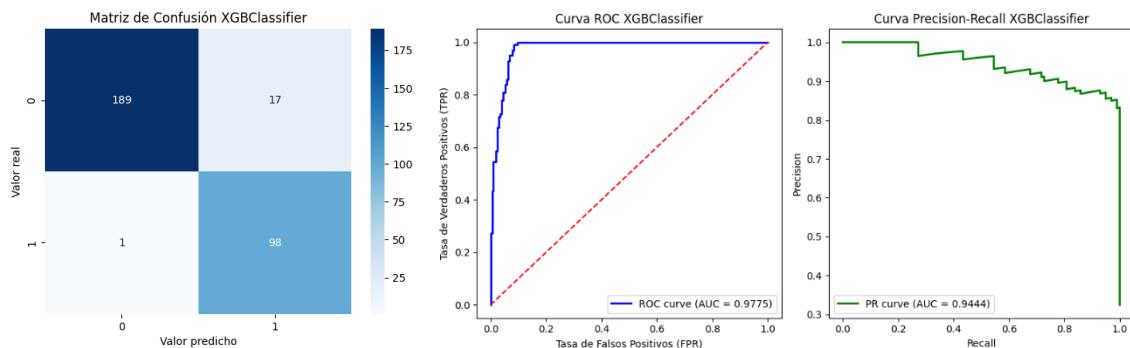


Figura 44: XGBoost: KNN - Matriz de confusión, curva ROC y curva PR.

#### MLP

- **Precisión:** 0,8190
- **Sensibilidad:** 0,9596

- **F1-Score:** 0,8837
- **AUC-ROC:** 0,9777
- **AUC-PR:** 0,9410

MLP tiene un comportamiento muy similar al SVM, con una alto sensibilidad (0,9596) y un f1-score competitivo (0,8837). Aunque es ligeramente mejor en sensibilidad que SVM, su precisión es muy similar. Su desempeño general es muy bueno, pero no supera a XGBoost en términos de f1-score ni precisión.

La curva ROC también es excelente, con un AUC de 0,9777, lo que coloca a este modelo cerca de los mejores clasificados.

La curva PR muestra una muy buena precisión, aunque decrece de manera más abrupta que los mejores modelos cuando la sensibilidad se acerca a 1. Esto indica que, aunque es muy bueno, podría tener más dificultades en los casos más difíciles.

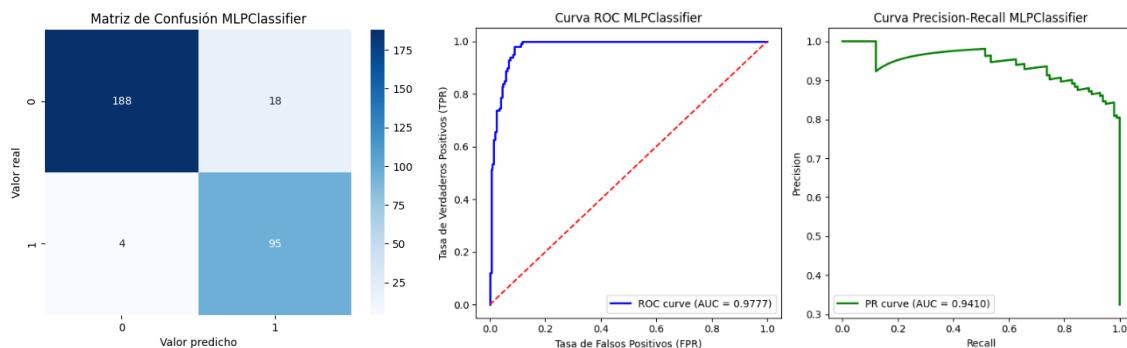


Figura 45: MLP - Matriz de confusión, curva ROC y curva PR

### Conclusiones

En base en las métricas presentadas, XGBoost se destaca como el mejor modelo por varias razones:

- Su sensibilidad es la más alta (0,9899), lo que significa que es el más eficaz en detectar casos de cáncer, minimizando los falsos negativos.
- Su precisión es superior a otros modelos como *Random Forest*, KNN y MLP, lo que indica que también tiene menos falsos positivos.
- El f1-score (0,9159), el AUC-ROC (0,9775) y el AUC-PR (0,9444) sugieren un excelente equilibrio entre precisión y sensibilidad.

En cuanto a el rendimiento de cada modelo, se concluye lo siguiente:

SVC y XGBoost parecen ser los modelos más robustos en términos de precisión, sensibilidad y F1-score, con curvas ROC y PR sobresalientes. Ambos mantienen alta precisión incluso con valores elevados de sensibilidad, lo que indica que son especialmente efectivos en la detección de casos de cáncer.

MLPClassifier sigue de cerca, mostrando una buena capacidad predictiva en general, aunque con una caída más notable en la precisión en valores altos de sensibilidad.



RandomForestClassifier también es un buen modelo, pero no supera a los dos modelos anteriores.

KNeighborsClassifier es el que muestra el rendimiento más débil entre los cinco modelos, con un AUC-PR menor y una caída en precisión más pronunciada.

Por tanto, XGBoost y SVC serían las mejores opciones, con una ligera ventaja para XGBoost debido a su mejor equilibrio entre las métricas y la capacidad de mantener alta precisión incluso en situaciones de alta sensibilidad.