



viu

**Universidad
Internacional
de Valencia**

Predicción de cáncer mediante modelos de Inteligencia Artificial aplicados a datasets de biopsias líquidas

Titulación:
Máster en Inteligencia
Artificial
Curso académico
2022 – 2023

Alumno/a: Cardells Tormo,
Ana
D.N.I: 44867476L

Director/a de TFM: Dr.
Ricardo Lebrón

Convocatoria:
Segunda

*What would your life be like
if you gave yourself permission to be
everything you wanted to be?*

Emilie Wapnick

Agradecimientos

Me gustaría expresar mi más sincero agradecimiento al Dr. Ricardo Lebrón, mi director del trabajo final de máster, por su dedicación, apoyo y guía durante todo el proceso de investigación.

También quisiera agradecer al profesorado del Máster en Inteligencia Artificial de la VIU por su valiosa enseñanza y contribución a mi formación académica.

Por último, pero no menos importante, quisiera agradecer a mi pareja Roberto Cano por su amor, comprensión y paciencia, que me han dado la fuerza y la motivación necesarias para completar este proyecto. Su apoyo incondicional ha sido fundamental en todo momento.

De nuevo, muchas gracias a todos los que han contribuido a la realización de este trabajo.

Índice general

Índice de figuras	III
Índice de tablas	v
Resumen	1
1. Introducción	3
1.1. Conceptos básicos	4
1.2. Estructura de los genes codificantes	5
1.2.1. Proceso de codificación de proteínas	5
1.3. Plaquetas Educadas por Tumores	8
2. Objetivos	11
3. Estado del arte	13
4. Metodología	15
4.1. Adquisición de los datos	15
4.2. Normalización de los datos	17
4.2.1. Normalización TPM	17
4.2.2. Cálculo de la longitud de cada tránscrito	19
4.2.3. Elección del tránscrito	19
4.3. Adición de etiquetas (<i>Ground Truth</i>)	21
4.4. Eliminación de valores atípicos	21
4.5. Partición de datos externa	24
4.6. Selección de características	24
4.7. Modelado	25
4.7.1. Selección de datos y etiquetas	25
4.7.2. Selección de métricas	26
4.7.3. Modelos de Soporte Vectorial	27
4.7.4. Modelos de bosque aleatorio (<i>random forest</i>)	29
4.7.5. Modelos de regresión logística	30



4.7.6. Modelos de aprendizaje profundo	32
4.7.7. Entrenamiento modelo definitivo	34
5. Resultados y Discusión	37
5.1. Resultados	37
5.1.1. Máquina de soporte vectorial (SVC)	37
5.1.2. Bosque aleatorio (<i>Random forest</i>)	38
5.1.3. Regresión Logística (LOGR)	39
5.1.4. Aprendizaje profundo	40
5.2. Discusión	41
6. Conclusiones	43
7. Limitaciones y Perspectivas de Futuro	45
7.1. Limitaciones	45
7.2. Perspectivas de Futuro	45
Lista de Acrónimos	47
A. Apéndice A	49
B. Apéndice B	51
B.1. Distribución de los datos	51
B.2. tSNE	51
B.3. PCA	52
B.4. rPCA	52
B.5. <i>Isolation Forest</i>	53
B.6. <i>Local Outlier Factor</i>	53
B.7. DBSCAN	54
B.8. Resultados	54
Bibliografía	55

Índice de figuras

1.1. Célula Eucariota	4
1.2. Estructura del ADN	5
1.3. Estructura del gen	6
1.4. Proceso de codificación de proteínas 1	7
1.5. Proceso de codificación de proteínas 2	8
1.6. Tabla del código genético	9
1.7. Estructura de las proteínas	10
1.8. Composición de la sangre	10
4.1. Distribución de edades	16
4.2. Distribución de género	16
4.3. Distribución de clase	17
4.4. Protocolo líquido	17
4.5. Obtención de las longitudes	20
4.6. tSNE sobre 9 genes	23
4.7. PCA sobre 9 genes	23
4.8. rPCA sobre 9 genes	24
4.9. Modelo de Soporte Vectorial	27
4.10. Modelo de bosque aleatorio	29
4.11. Modelo de regresión logística	31
4.12. Modelo de Red Neuronal	32
4.13. Sobreajuste en Redes Neuronales	34
4.14. Data Augmentation en Redes Neuronales, 1	35
4.15. Data Augmentation en Redes Neuronales, 2	35
4.16. Metodología	35
5.1. AUC SVC modelo 1	38
5.2. AUC Random Forest Modelo 1	39
5.3. AUC regresión logística Modelo 1	40
5.4. AUC Redes Neuronales	41

A.1. Primeras líneas archivo GFF	50
B.1. Distribución de datos de genes	51
B.2. Algoritmo t-SNE sobre los datos	52
B.3. Algoritmo PCA sobre los datos	52
B.4. Algoritmo rPCA sobre los datos	53

Índice de tablas

4.1. Búsqueda de valores atípicos usando panel de biomarcadores	22
4.2. Valores de clase	25
4.3. Hiperparámetros SVC	28
4.4. Hiperparámetros Random Forest	30
4.5. Hiperparámetros regresión logística	31
4.6. Hiperparámetros redes neuronales	34
5.1. Resultados SVC	37
5.2. Resultados Random Forest	38
5.3. Resultados regresión logística	39
5.4. Tiempos de ejecución en regresión logística	40
5.5. Resultados redes neuronales	40
5.6. Resumen de resultados	42
B.1. Búsqueda de valores atípicos usando todos los genes	54

Resumen

El **cáncer** es un conjunto de enfermedades caracterizadas por el crecimiento descontrolado de células anormales en el cuerpo, que pueden invadir y dañar tejidos y órganos cercanos. Estas células anormales también pueden diseminarse a otras partes del cuerpo a través del sistema linfático o sanguíneo, dando lugar a metástasis. El cáncer puede afectar a cualquier parte del cuerpo y su gravedad varía dependiendo del tipo de cáncer y la etapa en la que se encuentre al momento del diagnóstico.

Actualmente en España, los hombres tienen un riesgo de desarrollar cáncer antes de cumplir los 80 años de un 40,9 % y las mujeres de un 27,6 % según la Sociedad Española de Oncología Médica. Por tanto es muy importante la investigación orientada a la detección y tratamiento de estas enfermedades. Además, se sabe que la **detección temprana** permite tratamientos menos invasivos y una mayor probabilidad de superar la enfermedad.

Desde hace años se conoce que las **plaquetas**, aparte de su papel como coagulantes, también participan en la progresión del cáncer y la metástasis. Esto sucede una vez han entrado en contacto con células tumorales y han alterado la composición de su ARN mensajero (el principal mediador de la actividad de los genes). A este tipo de plaquetas alteradas se las conoce como **plaquetas educadas por tumores**. Las plaquetas se encuentran en la sangre en abundancia y pueden ser fácilmente extraídas (con un simple análisis de sangre) y aisladas. En los últimos años se han desarrollado modelos que permiten predecir la presencia del cáncer a partir del perfil de ARN mensajero presente en las plaquetas.

Para la elaboración de este trabajo, se ha decidido utilizar los **datos publicados en el artículo** [In't Veld et al. \(2022\)](#) para desarrollar modelos que permitan predecir de la mejor manera posible la presencia o ausencia de cáncer en un paciente a partir de una biopsia líquida (en este caso un análisis de sangre). Estos datos contienen información de 2351 pacientes repartidos entre pacientes enfermos de cáncer (69 %), clasificados por 18 tipos de cáncer diferente y pacientes sanos.

En la **preparación de los datos** utilizados en este estudio se emplearon diversas técnicas para obtener un conjunto de datos limpio y coherente. Para comenzar, las características (*features*) con las que se trabajó son cantidades que representan la captura de ARN mensajero de diferentes genes por parte de las plaquetas, habiendo una característica por gen. Como estas cantidades dependen de la cantidad de datos tomadas en cada paciente y de la longitud

de cada ARN mensajero, se utilizó una normalización propia de este campo del conocimiento conocida como tránscritos por millón o *transcripts per million* (TPM).

Además, se aplicaron técnicas de eliminación de datos anómalos mediante la utilización de diversos algoritmos de *clustering*, eliminando aquellos datos que fueron identificados como *outliers* por al menos dos de ellos. Después se hizo una selección de características, descartando aquellos genes con menos variabilidad o valores más bajos.

Una vez preparados los datos, se desarrollaron diferentes modelos de **aprendizaje supervisado y deep learning** para la predicción de la presencia o ausencia de cáncer en los pacientes. En los modelos de aprendizaje profundo se añadió en algunas de las ejecuciones *data augmentation* con el objetivo de reducir el *overfitting*. En la búsqueda de los hiperparámetros de los modelos, se aplicaron pesos distintos a cada clase para compensar el hecho de tener una base de datos desbalanceada. Además, se utilizaron técnicas como el *grid search* y *keras tuner* para agilizar la búsqueda y obtener los mejores resultados posibles. Finalmente se compararon los resultados obtenidos con los de artículo.

Durante el desarrollo de los modelos y su testeo, se puso mucho foco en obtener diferentes **métricas**. Una de las más importantes fue la especificidad. El objetivo fue intentar minimizar todo lo posible el número de falsos positivos, es decir personas a las que se les dijera que tenían un cáncer cuando en realidad estaban sanas.

Como **conclusión** mencionar que los modelos que mejor funcionaron fueron el de regresión logística y el de redes neuronales de dos capas ocultas con *data augmentation*. En ambos casos las mejores versiones de cada modelo fueron aquellas que tuvieron en cuenta el desbalanceo de las clases para añadir un peso mayor a la clase minoritaria. Como limitaciones mencionar que en este trabajo se realizó una clasificación binaria mientras que en el artículo se realiza también un clasificación multiclase para diferenciar entre los 18 tipos de tumores presentes en la base de datos. Si solo se tiene en cuenta la presencia o ausencia de cáncer, los resultados que se obtuvieron en este estudio son prometedores siendo ligeramente superiores a los del artículo.

Introducción

1

El **cáncer** es una enfermedad caracterizada por la proliferación incontrolada de células anormales en el cuerpo, que puede dar lugar a la formación de tumores malignos y a la invasión de tejidos y órganos cercanos. Esta enfermedad se produce cuando las células normales experimentan mutaciones en su material genético, que alteran los mecanismos que regulan su ciclo celular y su diferenciación. Como resultado, estas células anormales pueden replicarse y crecer a un ritmo acelerado, y tienen la capacidad de invadir los tejidos circundantes y propagarse a otras partes del cuerpo a través del sistema linfático o sanguíneo. El cáncer puede afectar a cualquier parte del cuerpo y puede manifestarse de diferentes maneras, dependiendo del tipo de célula y del tejido en el que se origina.

En la actualidad, el cáncer representa un gran desafío en la sociedad española, donde las tasas de incidencia continúan aumentando cada año. Según datos de la Sociedad Española de Oncología Médica, los hombres tienen una probabilidad del 40,9 % de desarrollar cáncer antes de cumplir los 80 años, mientras que para las mujeres esta probabilidad es del 27,6 %. Con este escenario en mente, se requiere una investigación orientada hacia la detección temprana y el tratamiento efectivo del cáncer.

Una herramienta prometedora para la **detección temprana** del cáncer es el uso de **plaquetas**, las cuales son conocidas por su papel en la coagulación de la sangre, pero también se ha descubierto que pueden desempeñar un papel clave en la progresión del cáncer y la metástasis. Cuando las plaquetas entran en contacto con células tumorales, modifican la composición del ARN mensajero en un proceso conocido como educación por tumores. Dado que las plaquetas se encuentran en gran cantidad en la sangre y son fácilmente extraíbles mediante un simple análisis de sangre, se ha comenzado a investigar la posibilidad de utilizar su perfil de ARN mensajero para predecir la presencia del cáncer.

La detección temprana del cáncer a través del **perfil de ARN mensajero de las plaquetas** no solo puede permitir tratamientos menos invasivos y aumentar la probabilidad de curación, sino que también puede ayudar en la identificación temprana de los pacientes que pueden estar en riesgo de desarrollar cáncer en el futuro. Por lo tanto, el uso de plaquetas como herramienta de detección temprana es un área de investigación prometedora en la lucha contra el cáncer.

A continuación se explican conceptos de biología y genética que serán necesarios para entender el resto del contenido de este trabajo. La mayoría del material ha sido extraído de los siguientes libros: [Copelli \(2010\)](#) y [Rodríguez Arnaiz et al. \(2016\)](#).

1.1. Conceptos básicos

Los seres humanos son organismos **eucariontes** y por ello sus células (eucariotas) presentan un núcleo definido que está separado del resto de la célula por una doble membrana. Este se comunica con el resto de la célula a través de unos poros y su actividad está altamente regulada (véase Figura 1.1).

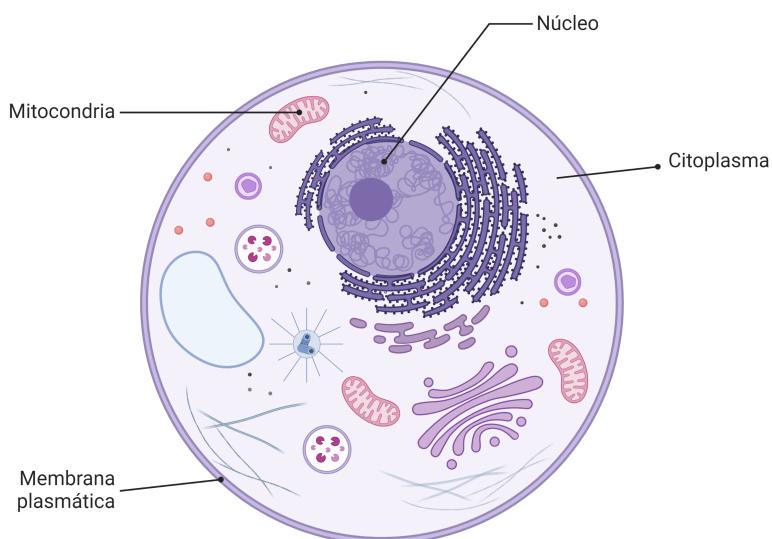


Figura 1.1: Célula Eucariota (creado con BioRender.com)

El **genoma** contiene toda la información necesaria para el crecimiento y el desarrollo de una persona. Se encuentra en su mayoría en el núcleo de la célula en forma de ácido desoxirribonucleico (ADN) aunque también en el interior de las mitocondrias.

En el núcleo de la célula, el genoma se divide en **cromosomas**. Un cromosoma es una molécula muy larga de ADN de doble hélice (bicatenario) independiente de otras moléculas de ADN bicatenario (es decir, otros cromosomas). En los seres humanos, normalmente cada célula contiene 23 pares de cromosomas (un total de 46). El ADN está compuesto de 4 bases nitrogenadas: Adenina (A), Citosina (C), Guanina (G) y Timina (T). Las bases de una y otra hebra están emparejadas, siendo las posibles parejas: A con T y G con C (véase Figura 1.2).

Un **gen** es un segmento definido de ADN que se encuentra dentro del cromosoma, ubicado siempre en una posición específica denominada *locus*. Es la unidad mínima de información genética funcional. Los genes más estudiados son los que tienen como producto funcional una proteína, también conocidos como **genes codificantes** y en ellos se centrará el presente trabajo.

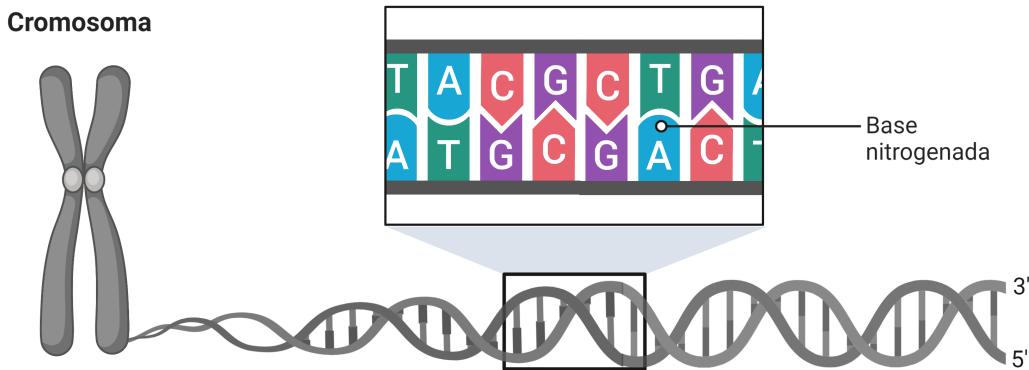


Figura 1.2: Estructura del ADN (creado con BioRender.com)

1.2. Estructura de los genes codificantes

Los genes codificantes se componen de las siguientes partes (véase Figura 1.3):

- **Promotor:** es una región de ADN que controla la iniciación de la transcripción de una determinada porción del ADN a ARN. La región promotora está compuesta por una secuencia específica de ADN localizada justo donde se encuentra el punto de inicio de la transcripción del ADN y contiene la información necesaria para activar o desactivar el gen que regula.
- **5' UTR :** 5' (leído cinco prima) región no traducida o *untranslated region* (UTR) es el sector extremo del ARN mensajero. En este caso la región no traducida 5' (5'UTR) marca el inicio. Se encuentra colindando con el marco de lectura abierto.
- **Exones:** es la parte del gen que contiene el ADN codificante, es decir, que permite sintetizar proteínas.
- **Intrones:** es la parte del gen que contiene ADN no codificante. Se elimina durante la maduración o splicing del ARN mensajero.
- **3' UTR:** Marca el final del ARN mensajero.
- **Terminador:** es una región de ADN que marca el final de la transcripción de un gen.

Un gen puede contener distinto número de exones e intrones.

1.2.1. Proceso de codificación de proteínas

El proceso de síntesis de una proteína es algo complejo, pero a continuación se explican los aspectos más importantes:

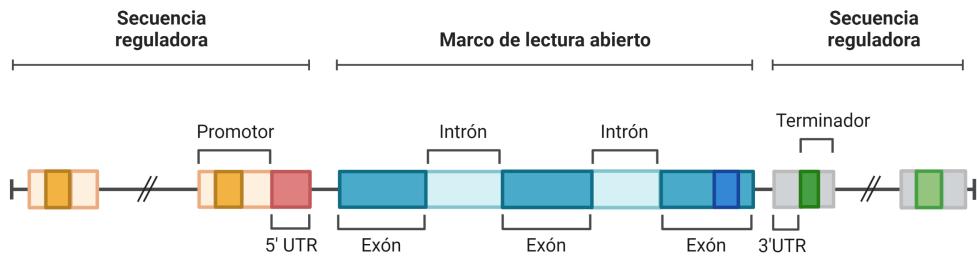


Figura 1.3: Estructura del gen (creado con BioRender.com)

- **Transcripción:** Transcribir (o copiar) el ADN en forma de ácido ribonucleico (ARN) mensajero (ARNm). La cadena de ARN es complementaria a la plantilla del ADN con la excepción de que las tiaminas (T) se reemplazan por uracilos (U) en el caso del ARN. Además, el ARNm se transporta desde el núcleo de la célula al citoplasma. Véase Figura 1.4.
- **Procesado del ARN mensajero (ARNm):** Eliminar aquellas partes que no aportan información sobre la síntesis de proteínas (intrones) hasta obtener el ARN mensajero maduro. Véase Figura 1.4. La información se encuentra en el ARNm en forma de tripletes o **codones**. Los codones están constituidos por tres **nucleótidos** consecutivos y no solapantes. En este documento se usarán los conceptos bases nitrogenadas y nucleótidos como sinónimos, aunque realmente la base nitrogenada es una parte del nucleótido.
- **Traducción:** A partir de la información existente se sintetiza, de codón en codón, una cadena de aminoácidos. Véase Figura 1.5. La correspondencia entre cada secuencia de nucleótidos (codón) y el aminoácido correspondiente la establece el llamado "código genético" (no confundir con información genética), el cual se muestra en la Figura 1.6. Por ejemplo, UUU y UUC corresponden al aminoácido Phe (Fenilalanina) que se encuentra en las proteínas como L-fenilalanina (LFA), siendo uno de los nueve aminoácidos esenciales para el ser humano. Esta cadena de aminoácidos finalmente se pliega tridimensionalmente dando lugar a la proteína.

En la figura 1.7 muestra los diferentes niveles de organización que conforman la **estructura de una proteína**:

- Estructura primaria: Este nivel de organización se refiere a la secuencia lineal de aminoácidos que conforman la proteína. Cualquier cambio en la secuencia primaria, incluso uno solo de los aminoácidos, puede alterar significativamente la estructura y función de la proteína.
- Estructura secundaria: Este nivel de organización se refiere a la forma en la que la proteína se pliega localmente, es decir, la disposición tridimensional de pequeñas

regiones de la cadena polipeptídica. Existen tres tipos principales de estructuras secundarias: alfa-hélices, beta-láminas plegadas y giros.

- Estructura terciaria: Este nivel de organización se refiere al plegamiento de la estructura secundaria en una estructura tridimensional completa y funcional. El plegamiento correcto es esencial para que la proteína pueda realizar su función biológica.
- Estructura cuaternaria: Este nivel de organización se refiere a la estructura de orden superior formada por la interacción de varias cadenas polipeptídicas. Algunas proteínas están formadas por una sola cadena polipeptídica y por lo tanto no tienen una estructura cuaternaria. Sin embargo, muchas proteínas están formadas por varias cadenas polipeptídicas que se unen para formar una estructura más compleja y funcional.

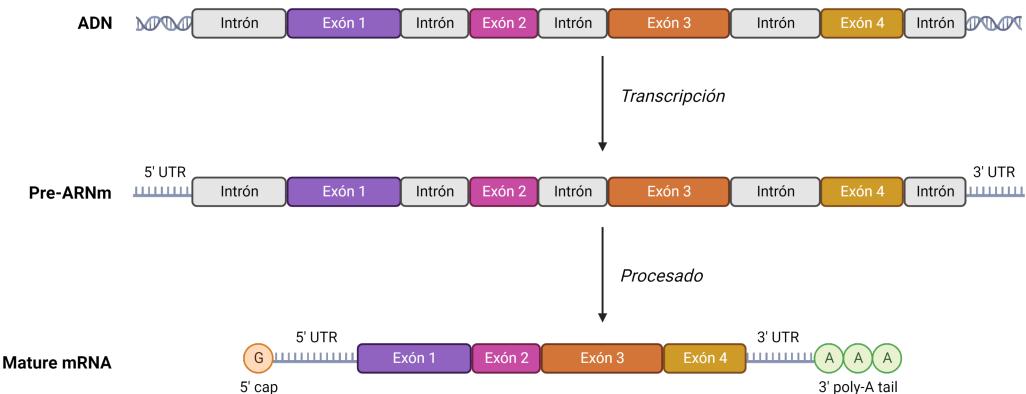


Figura 1.4: Proceso de codificación de proteínas: Transcripción y Procesado (creado con BioRender.com)

La cantidad de **ARN mensajero** de un gen es un indicativo de cuánto de activo está dicho gen. Es decir, cuanto más ARN mensajero se encuentre, más activo está el gen correspondiente, existiendo una relación directamente proporcional entre la cantidad de ARN mensajero y la actividad del gen.

Algunos genes sólo están activos en un tipo de célula, por lo que serán buenos indicadores del origen de un tumor originado a partir de dicho tipo celular. Otros, en cambio, están activos en dos o más tipos de células, siendo el caso extremo el de los genes *housekeeping*, necesarios para la supervivencia de cualquier célula humana.

Además, algunos genes sólo se activan durante el desarrollo embrionario o en determinadas etapas de la vida (por ejemplo, infancia o vejez). Otros genes se activan en respuesta a estímulos ambientales (por ejemplo, defensa contra la luz ultravioleta) o cambian su actividad a lo largo del día o de las estaciones, marcando ritmos biológicos como el sueño.

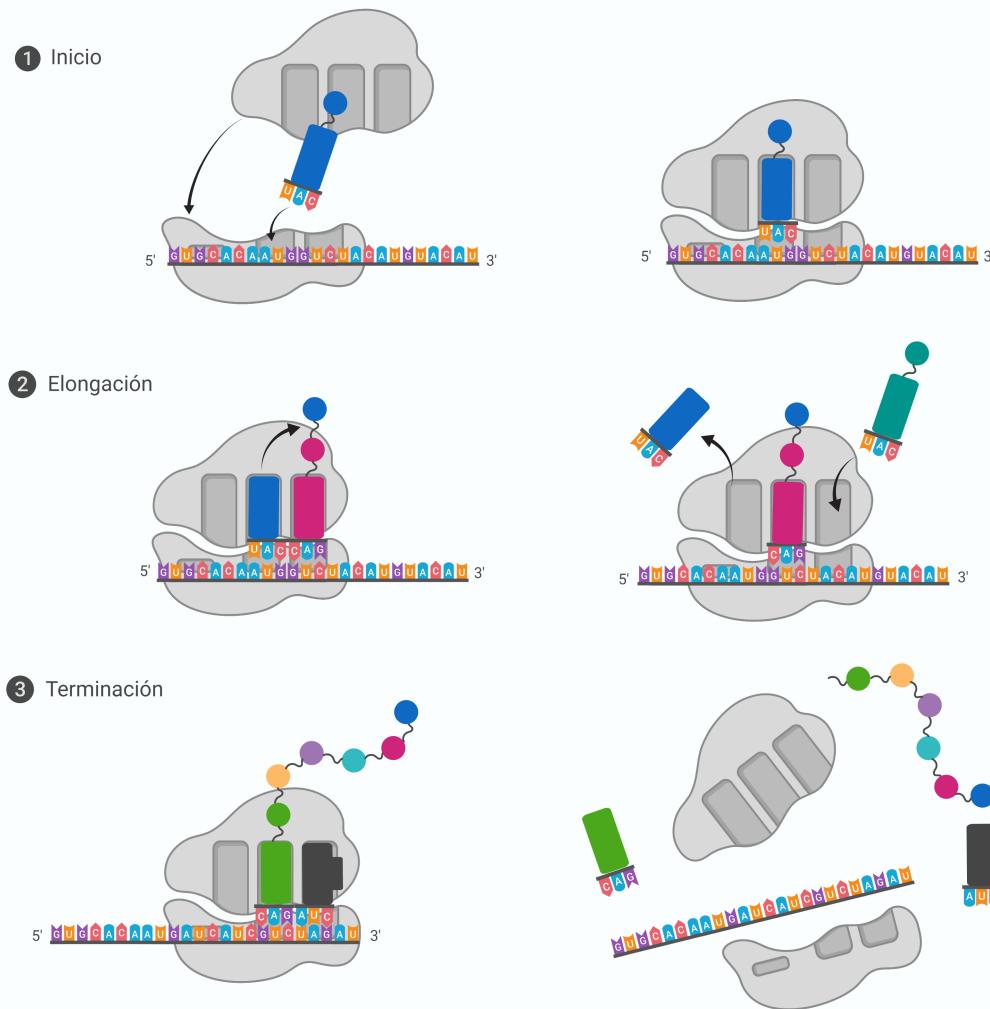


Figura 1.5: Proceso de codificación de proteínas: Traducción del ARN mensajero a proteína (creado con BioRender.com)

1.3. Plaquetas Educadas por Tumores

La **sangre** (véase Figura 1.8) es un tipo especial de tejido líquido formado por:

- **Plasma:** Es el medio líquido en que viajan el resto de componentes de la sangre y está compuesto por agua, sales y proteínas.
- **Parte sólida:** Contiene mayoritariamente glóbulos rojos, glóbulos blancos y plaquetas.

Las **plaquetas** son fragmentos de otras células más grandes llamadas megacariocitos. Por ese motivo no tienen un núcleo completo y activo (no son capaces de generar ARNm por sí mismas), aunque sí contienen ARN mensajero procedente de los megacariocitos y adquirido por contacto con otras células en su recorrido por el organismo a través de la sangre. La función principal de las plaquetas es contribuir a la formación de coágulos en respuesta a lesiones.

Segunda base del codón				Última base del codón	
	U	C	A		
U	UUU Phe UUC UUA Leu UUG	UCU Ser UCC UCA UCG	UAU Tyr UAC UAA STOP UAG	UGU Cys UGC UGA STOP UGG Trp	Última base del codón
C	CUU Leu CUC CUA CUG	CCU Pro CCC CCA CCG	CAU His CAC CAA Gln CAG	CGU Arg CGC CGA CGG	
A	AUU Ile AUC AUA AUG Met (start)	ACU Thr ACC ACA ACG	AAU Asn AAC AAA Lys AAG	AGU Ser AGC AGA Arg AGG	Última base del codón
G	GUU Val GUC GUA GUG	GCU Ala GCC GCA GCG	GAU Asp GAC GAA Glu GAG	GGU Gly GGC GGA GGG	

Figura 1.6: **Tabla del código genético** (creado con BioRender.com)

Sin embargo, como se explica en el artículo [Varkey y Nicolaides \(2021\)](#), se sabe que cuando las plaquetas interaccionan con células tumorales, éstas se convierten en **plaquetas educadas por tumores o tumor-educated platelets (TEP)** y su perfil de ARN mensajero cambia. De esta forma, medir y analizar las cantidades de ARNm en las plaquetas en sangre mediante una **biopsia líquida** abre la puerta a potenciales usos como: diagnóstico, predicción o seguimiento de diferentes tipos de cáncer.

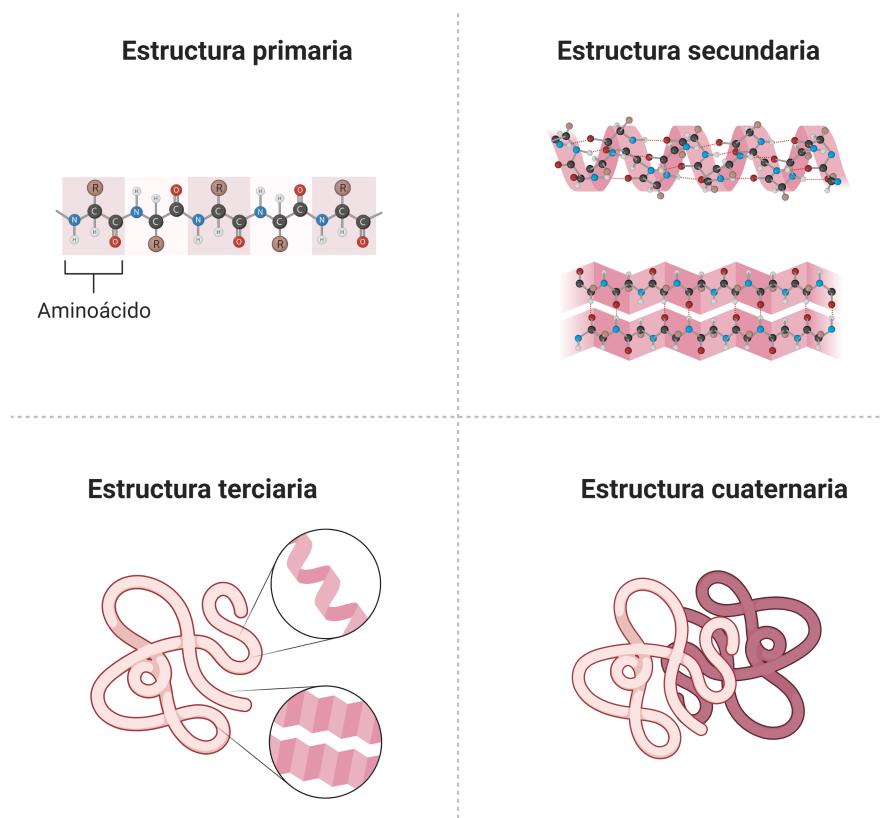


Figura 1.7: **Estructura de las proteínas** (creado con BioRender.com)

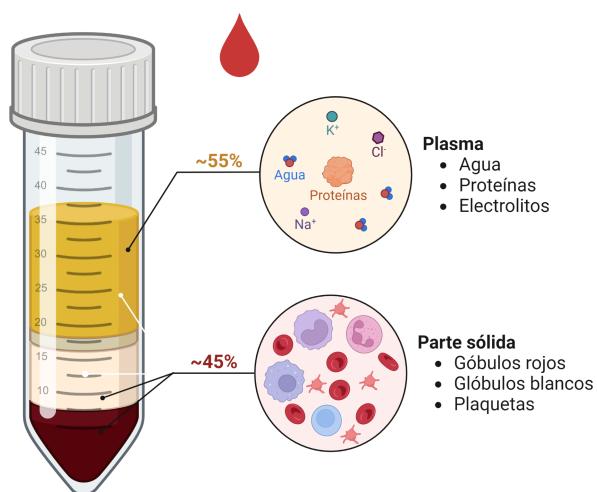


Figura 1.8: **Composición de la sangre** (creado con BioRender.com)

Objetivos

2

El objetivo principal del trabajo fue desarrollar un modelo de inteligencia artificial que permitiera predecir de manera precisa la presencia o ausencia de cáncer en pacientes a partir de una biopsia líquida, en este caso un análisis de sangre, utilizando los perfiles de ARN mensajero presentes en las plaquetas educadas por tumores de la sangre.

Para ello, el trabajo se dividió en los siguientes objetivos secundarios:

- 1. Paliar el efecto de trabajar con una base de datos desbalanceada:** La base de datos con la que se trabajó está desbalanceada, presentando un 69 % de etiquetas de valor "1"(personas enfermas). Se tenía la hipótesis de que esta desigualdad en la distribución de las etiquetas de la base de datos podía llevar a que algunos modelos presentaran una exactitud muy alta, pero una baja especificidad. Por ese motivo se decidió explorar técnicas de balanceo de datos y comparar los resultados entrenando modelos con y sin estas técnicas. Entre las herramientas que se utilizaron está el uso de pesos por clase, donde se asigna un peso mayor a la clase menos representada, o la sobrecolección de la clase minoritaria con el algoritmo SMOTE (*Synthetic Minority Over-sampling Technique*) donde se generan datos sintéticos de la clase minoritaria hasta obtener el número de muestras deseado.
- 2. Obtener métricas representativas del problema:** Se decidió trabajar con una conjunto de métricas y medir cada modelo en base a ellas: exactitud (*accuracy*), área bajo la curva ROC (*auc*), sensibilidad y especificidad. En el artículo [In't Veld et al. \(2022\)](#) se menciona esta última como una métrica crítica ya que maximizarla implica minimizar el número de falsos positivos, algo muy importante cuando se trabaja con personas. Se deseaba compararlas y ver el efecto de maximizar una de ellas en el resto de métricas.
- 3. Comparar diferentes subconjuntos de las características:** Se quería comprobar si había una diferencia significativa entre utilizar todas las características, solo una parte o un número reducido de características (usando un panel de diez biomarcadores que se menciona en el artículo).
- 4. Comparar diferentes modelos:** Se decidió utilizar tres modelos de clasificación de aprendizaje supervisado: máquina de soporte vectorial (que es el estado del arte actual), bos-

que aleatorio y regresión logística. También se probaron modelos de aprendizaje profundo (*deep learning*), con una red neuronal con dos capas densas ocultas, *drop out* y *batch normalization*. El objetivo era conocer el diferente grado de desempeño de cada una de ellas. También se quería conocer si el hecho de tener pocos datos podía ser un impedimento para el uso de redes neuronales o si por el contrario serían más capaces de encontrar patrones que los algoritmos de aprendizaje supervisado.

5. **Comparar el modelo propuesto con el estado del arte actual:** Por último se quería comparar el resultado de los modelos creados con los resultados que se obtienen en el artículo.

Estado del arte

3

La detección de cáncer mediante el estudio de plaquetas educadas por tumores (TEP) está ganando terreno en la investigación médica. En particular, se han desarrollado protocolos de secuenciación de ARN mensajero (ARNm) a partir de TEP que permiten obtener información sobre la expresión génica de las plaquetas y su relación con el cáncer. Por ejemplo, en el estudio de [Best et al. \(2019\)](#) se presenta un protocolo para el aislamiento de ARNm de TEP y un algoritmo de clasificación basado en SVM que logra una exactitud del 88 % en la detección de gliomas de bajo grado. Además, otros estudios han aplicado este enfoque para la detección de diferentes tipos de cáncer, como el cáncer de endometrio ([Łukasiewicz et al., 2021](#)), el carcinoma de células renales ([Xiao et al., 2022](#)) y el carcinoma de células escamosas de esófago ([Liu et al., 2022](#)).

Para lograr una mayor precisión en la detección de cáncer, los algoritmos de clasificación basados en inteligencia artificial se han convertido en una herramienta importante en el procesamiento de los datos de secuenciación. Estos algoritmos se encargan de analizar los patrones de expresión génica de las plaquetas y determinar si hay una relación con la presencia de cáncer. En algunos estudios, como el de [In't Veld et al. \(2022\)](#), se han utilizado múltiples algoritmos de clasificación, principalmente SVM, para detectar diferentes tipos de cáncer. Los resultados obtenidos en estos estudios son prometedores y sugieren que el análisis de TEP puede ser una herramienta útil en la detección temprana de cáncer y en la monitorización de la progresión de la enfermedad.

A continuación se resume el contenido de alguno de los artículos consultados:

1. En el artículo [Best et al. \(2019\)](#) se presenta un protocolo para obtener librerías de ARN mensajero (ARNm) secuenciado a partir de plaquetas educadas por tumores o *tumor-educated platelets* (TEP). Este protocolo incluye los pasos:

- Aislamiento de las plaquetas del resto de elementos de la sangre
- Aislamiento de ARNm proveniente de las plaquetas
- Secuenciación de ARN mensajero

2. En ese mismo artículo se describe un protocolo para el desarrollo de algoritmos de clasificación basados en inteligencia artificial. Este protocolo incluye los siguientes pasos:

- Preprocesado de las base de datos

-
- Controles de calidad de los datos
 - Normalización de los datos
 - Modelado. En este caso los autores hicieron uso del algoritmo *support vector machine* (SVM)

Obtuvieron una exactitud (*accuracy*) de un 88 % en el algoritmo de clasificación para la detección del glioma de bajo grado (un tipo de tumor cerebral).

3. En Łukasiewicz et al. (2021) se presenta un estudio sobre la utilidad de usar TEP y ADN de tumores circulante para realizar un diagnóstico de cáncer de endometrio. Respecto al estudio con TEP, los autores aplicaron un algoritmo de clasificación y obtuvieron los siguientes resultados: *Cross-validated area under curve* (AUC) del 97.5 % cuando se discrimina entre pacientes sanos y pacientes con cáncer. Este número bajó al 84.1 % si se pretende discriminar pacientes con cáncer de pacientes con condiciones ginecológicas benignas.
4. En Xiao et al. (2022) se plantea el uso de TEP para detección del carcinoma (tipo de cáncer que surgen de órganos que tienen algún epitelio como el tubo digestivo, la piel o las mucosas) de células renales mediante modelos de clasificación basados en SVM. Los autores obtuvieron una *accuracy* del 88.9 % (AUC: 0.963) en validación tras la selección de los 68 genes biomarcadores que mejor contribuyeron al modelo.
5. En Liu et al. (2022) también se utiliza secuenciación de ARNm proveniente de TEP y un algoritmo de clasificación tipo SVM para la detección del carcinoma de células escamosas de esófago. En este caso se seleccionan únicamente tres genes: ARID1A, GTF2H2 y PRKRIR. Los resultados fueron una sensibilidad del 87.5 %, una especificidad del 81.1 % y una AUC de 0.893 separando pacientes sanos de pacientes con este tipo de cáncer.
6. En Int' Veld et al. (2022) se plantea el uso de ARNm de TEP para detectar 18 tipos de cancer mediante algoritmos de clasificación basados en inteligencia artificial (principalmente SVM). Los autores obtuvieron resultados del 99 % de especificidad si solo se incluyen controles asintomáticos o *asymptomatic controls* (ACs) y 78 % de especificidad si también se incluyen controles sintomáticos o *symptomatic controls* (SCs).

Metodología

4

La metodología empleada en este estudio constó de varias etapas interconectadas, cuyo objetivo fue desarrollar un modelo de detección de cáncer preciso y eficiente mediante el uso de técnicas de aprendizaje supervisado y aprendizaje profundo. En primer lugar, se llevó a cabo la obtención y preprocesamiento de los datos necesarios para el estudio (normalización y eliminación de datos anómalos). A continuación, se procedió a la selección de características relevantes mediante técnicas de análisis estadístico. En tercer lugar, se desarrollaron modelos de aprendizaje supervisado, basados en algoritmos clásicos como SVM o regresión logística, y modelos de aprendizaje profundo, basados en redes neuronales. En esta sección se describirá en detalle cada una de estas etapas, así como las herramientas y técnicas específicas que se utilizaron en cada una de ellas.

Se puede encontrar todo el código desarrollado para este trabajo en este repositorio de GitHub: <https://github.com/anacardells/TFM>.

4.1. Adquisición de los datos

Para la elaboración de este trabajo, se utilizaron los datos compartidos en el artículo [In't Veld et al. \(2022\)](#). En él se recolectan biopsias líquidas de 2.351 individuos con las siguientes características:

- Edades: entre 18 y 92 años (véase Figura 4.1)
 - Género: hombres y mujeres (véase Figura 4.2)
 - Procedencia: Europa y Norte América
 - Estado de salud (véase Figura 4.3):
 - 723 personas sanas
 - 390 controles asintomáticos o *asymptomatic controls* (ACs)
 - 333 controles sintomáticos o *symptomatic controls* (SCs): incluyendo enfermedades cardiovasculares, masas benignas o condiciones inflamatorias entre otras, pero sin diagnóstico de cáncer
 - 1628 personas enfermas con cáncer (18 tipos de cáncer diferentes identificados).
- En el presente trabajo se decidió trabajar con un problema binario (enfermo/sano)

para simplificar el proceso de modelado, reducir el tiempo de entrenamiento y mejorar la precisión de la identificación de pacientes con cáncer

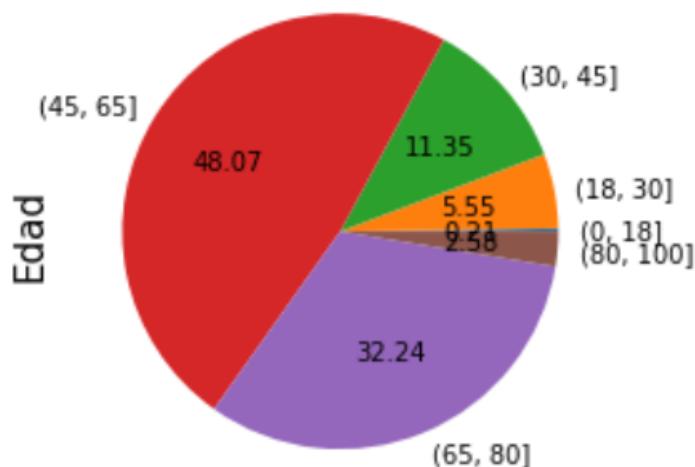


Figura 4.1: Distribución de edades, del artículo [In't Veld et al. \(2022\)](#)

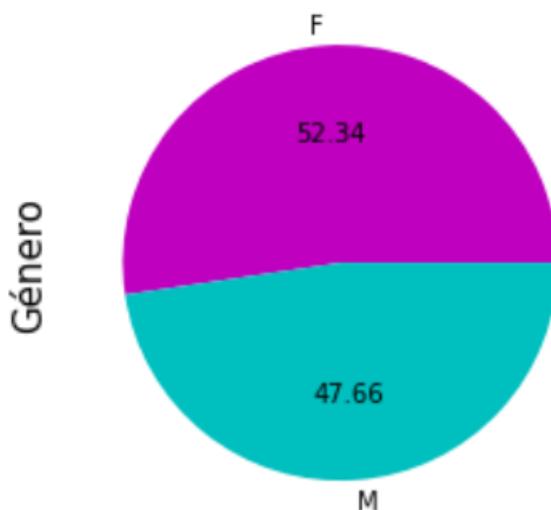


Figura 4.2: Distribución de género, del artículo [In't Veld et al. \(2022\)](#)

Se indica en el artículo que las muestras de sangre fueron tomadas en el momento del diagnóstico o durante el tratamiento y que se siguió el protocolo que se presenta en el artículo [Best et al. \(2019\)](#). Este protocolo se explica en el apartado 3. Se puede ver un resumen gráfico en la Figura 4.4. El resultado es un listado con 5440 genes (características) por paciente.

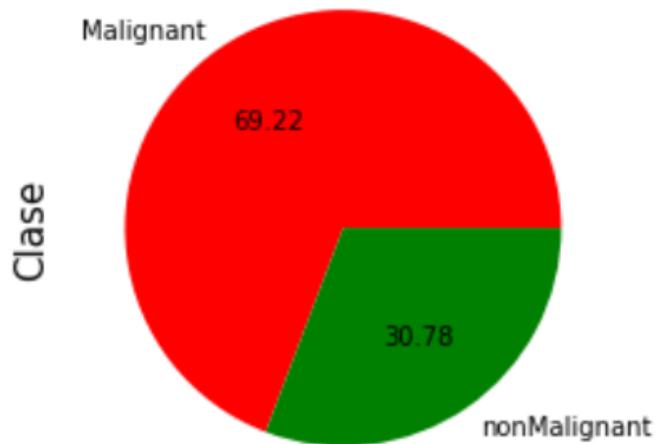


Figura 4.3: *Distribución de clase*, del artículo [In't Veld et al. \(2022\)](#)

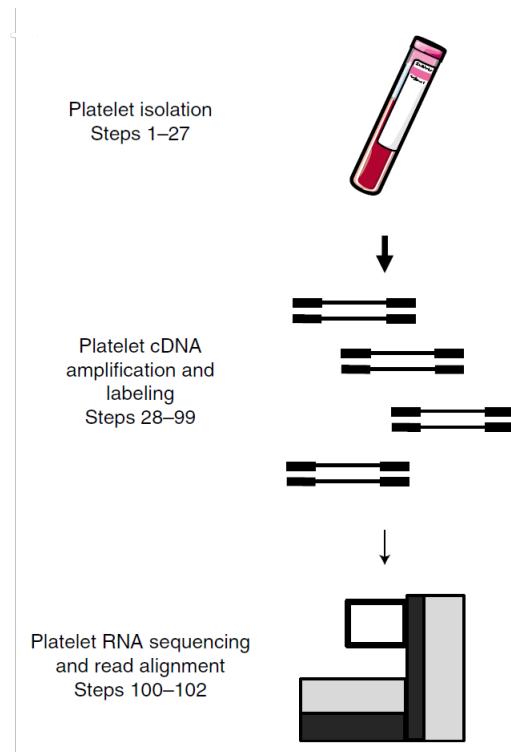


Figura 4.4: *Protocolo líquido*, del artículo [Best et al. \(2019\)](#)

4.2. Normalización de los datos

4.2.1. Normalización TPM

Cuando se trabaja con bases de datos de secuenciación de genes, se utilizan normalizaciones específicas de este dominio. En concreto, una de las más comunes y la que se utilizó

en este trabajo es la normalización tránscritos por millón o *transcripts per million* (TPM).

Un **tránscribo** es la molécula de ARN monocatenario que se obtiene inmediatamente después de la transcripción (es decir, que está formada por la unión de intrones y exones). Como ya se explicó en 1.2.1, estos tránscritos primarios requieren de modificación post transcripcional para la obtención de ARN maduro y plenamente funcional. En este proceso de maduración se eliminan los intrones.

TPM se introdujo para facilitar las comparaciones entre muestras y permite corregir dos sesgos que se suelen dar en este tipo de experimentos:

- **Longitud del gen:** Si el gen A es más largo que el gen B, esto significa que probablemente se van a encontrar más fragmentos del primero. Cuando se cuantifican y mapean estos fragmentos a los genes correspondientes, el número de fragmentos detectados del ARN mensajero (*counts*) del gen A será probablemente superior al gen B. Por lo tanto, si únicamente se compara el número de apariciones, parecería que el gen A está siendo más expresado que el gen B, lo cual no es verdad.
- **Profundidad de la secuenciación:** La profundidad de secuenciación en la normalización TPM hace alusión al número total de fragmentos secuenciados para una instancia dada (observación o paciente). Este valor cambia de observación en observación y está directamente relacionado con la probabilidad de observar fragmentos de cada ARN mensajero. Si la muestra 1 tiene una mayor profundidad que la muestra 2, se corrige de manera que todas las muestras pesen lo mismo (en este caso todas las lecturas de una muestra suman un millón).

La fórmula a aplicar para calcular el TPM es:

$$TPM_i = \frac{q_i/l_i^{Kb}}{\sum_j(q_j/l_j^{Kb})} * 10^6$$

Donde q_i denota las lecturas por cada tránscrito, l_i^{Kb} la longitud de cada tránscrito en Kilobases (Kb) y el sumatorio del denominador corresponde a la suma de todos los tránscritos de una muestra normalizados por su longitud. El sumatorio de TPM_i es siempre un millón.

Como se puede comprobar, este cálculo se realiza independientemente para cada muestra (su resultado no depende de otras muestras) y por tanto se puede realizar antes de la partición entre *train* y *test*.

En el caso de la base de datos utilizada se encontró un obstáculo para poder aplicar la normalización TPM: En la base de datos de pacientes, se da información de cantidades (*counts*) de genes pero no de tránscritos.

Además, se tuvo que realizar un paso adicional para obtener la información de la longitud de los tránscritos.

Estos dos pasos se resolvieron como se detalla en las siguientes sub-secciones.

4.2.2. Cálculo de la longitud de cada tránscrito

En el artículo del que se obtuvo la base de datos, se indica que se ha trabajado con la versión del genoma *human genome 19* (HG19). HG19 es a su vez el alias para *Human Gene Annotation - GENCODE Release 19* (GRCh37) que se puede descargar desde la página web https://www.gencodegenes.org/human/release_19.html indicada en el artículo [Frankish et al. \(2019\)](#). En concreto se descarga la versión en formato GFF. El formato *General Feature Format* (GFF) es un formato de archivo utilizado en genómica y bioinformática para describir las características y anotaciones de los genes y otros elementos genómicos en un genoma secuenciado. Véase apéndice A para más información.

Una vez descargada la notación génica, se apreció que:

- La notación génica es una lista de coordenadas de distintos elementos genómicos, entre ellos genes, tránscritos y sus componentes
- Para cada gen se detallan una lista de tránscritos (o isoformas). Es importante notar que la transcripción de un gen puede implicar a grupos diferentes de exones, resultando en distintas proteínas
- Para cada tránscrito, se detallan sus componentes (intrones, exones...) y la posición de cada uno de ellos

Tal y como se mencionó en las secciones [1.2](#) y [1.2.1](#), para calcular la longitud de un tránscrito (en el caso del estudio de ARN mensajero) bastará con sumar las longitudes de cada uno de los exones que lo componen (ya que en el ARN mensajero solo están presentes los exones). Para calcular la longitud de los exones (en número de bases), bastará con aplicar la siguiente fórmula:

$$l_i = pos_{final} - pos_{ini} + 1$$

Se ha de sumar 1 ya que las posiciones de inicio y final son inclusivas.

4.2.3. Elección del tránscrito

En la base de datos de pacientes, se da información de cantidades (*counts*) de genes pero no de tránscritos. Cada gen se puede expresar con más de una isoforma (más de un tránscrito) y no se conoce cuál ha sido medida.

Primero, se revisó el listado generado en [4.2.2](#), pero no se pudo utilizar directamente ya que cada gen de la base de datos tenía entre 1 y 82 tránscritos (con una media de 3.4 tránscritos) y en este caso se necesita una relación unívoca entre el nombre del gen y el nombre del tránscrito para poder aplicar la normalización TPM.

Para salvar este punto se acudió a la página de USCS genome <http://genome.ucsc.edu> y se descargó un listado con la **isoforma canónica** (el mejor tránscrito) para cada gen (<https://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/knownCanonical.txt.gz>). Se indica que generalmente se trata de la isoforma más larga, pero no siempre.

Los genes de este listado se encuentran en formato University of California Santa Cruz (UCSC) y han de ser mapeados al formato GENCODE. Para ello de <https://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/knownToEnsembl.txt.gz> se descargó la tabla *knownToEnsembl* y se realizó la equivalencia.

A continuación se revisaron los 5440 genes de la base de datos de pacientes y se observó que:

- 5251 Genes (96.5 %) cuentan con una única isoforma en la tabla *knownCanonical* y por tanto se puede utilizar la longitud de esta isoforma.
- 98 Genes cuentan con más de una isoforma en la tabla *knownCanonical*.
- 91 Genes no aparecen en la tabla *knownCanonical*.

No era posible conocer qué estrategia se usó para la elaboración del artículo. En este momento se presentaban dos opciones: utilizar únicamente los 5251 genes que tienen una relación unívoca en *BestTranscripts* o usar siempre el tránscrito más largo para cada uno de los 5440 genes de la base de datos. En este caso se decidió utilizar la primera estrategia, considerando que la información se distribuye en la red génica y no parecía probable que la información útil para clasificar los pacientes se concentrara en esos 98 + 91 genes que se iban a eliminar. Véase Figura 4.5.

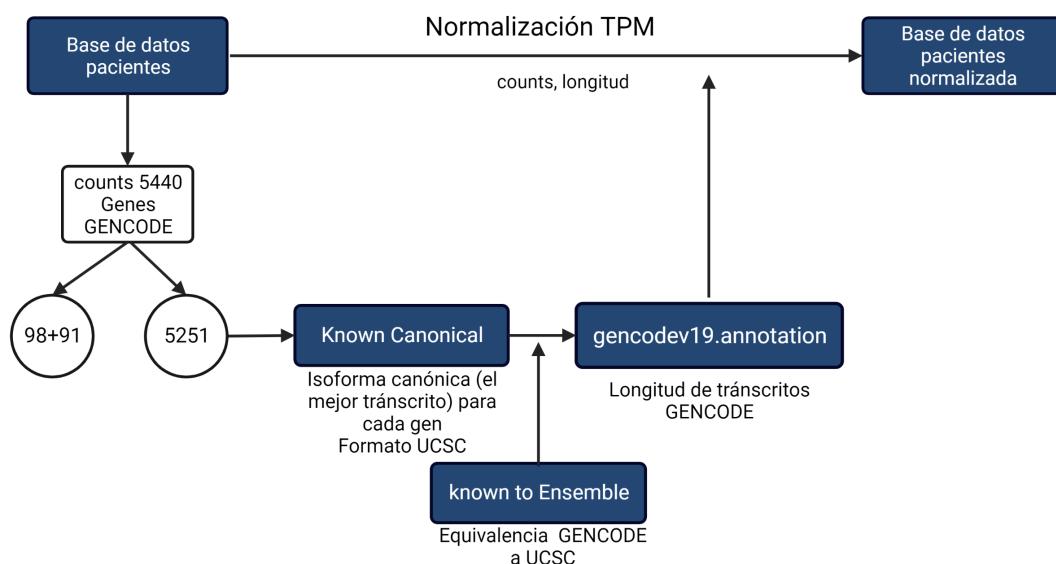


Figura 4.5: Obtención de las longitudes de los tránscritos para aplicar TPM (creado con BioRender.com)

4.3. Adición de etiquetas (*Ground Truth*)

La base de datos que se ha utilizado lleva asociados dos archivos de *Ground Truth* (GT) o etiquetas y características adicionales: uno con formato txt y otro con formato xml.

Para consolidar la información de ambos archivos y verificar la información de los mismos, se realizaron las siguientes acciones:

- Cambiar los nombres de algunas etiquetas para que coincidan. Por ejemplo se cambió la etiqueta *Pancreatic Disease* por *Pancreatic diseases*.
- Revisar la información y corregir aquellos campos que presentaban errores. Por ejemplo:
 - Dentro de la clase *Malignant* existían veintiún pacientes marcados como *Asymptomatic controls*.
 - Dentro de la clase *nonMalignant* existía un paciente marcado como *Prostate Cancer*
 - Dentro de la clase *nonMalignant*, existía un paciente que tenía el valor IV en la columna *Stage*

Una vez contrastados los nuevos números corregidos con los datos que se presentan en el artículo, todas las clases y grupos coincidían: 723 personas sanas (de las cuales 390 son AC y 333 son SC) y 1628 personas con cáncer.

- Se tomó nota de que las columnas *Stage*, *Sex* y *Age* presentaban algunos datos nulos (algunos pacientes carecían de esa información)

Para poder fusionar estos archivos con el archivo que contenía los datos normalizados (proveniente de [4.2](#)), se revisó el nombre codificado de los pacientes en ambos:

- En 2348 pacientes el nombre era casi idéntico, si se eliminaba el cardinal que precede al nombre en uno de los archivos. Por ejemplo, se podía entender que *1-Vumc-HD-101-TR922* y *Vumc-HD-101-TR922* representaban al mismo paciente.
- En tres casos, los nombres no eran tan parecidos, aunque aún se encontraban ciertas similitudes. Por ejemplo: *countMatrix.506-NKI-NSCLC-107-270* y *NKI-NSCLC-107-270-TR725*. Se asumió que en cada pareja, se trataba del mismo paciente.

Tras este análisis, se asumió que ambas bases de datos tenían la misma lista de pacientes y que, además, estaban en el mismo orden y por tanto se procedió a fusionar ambas.

4.4. Eliminación de valores atípicos

La primera decisión a tomar era si se deseaba eliminar valores atípicos antes o después de realizar la partición de datos externa. Ambas estrategias presentaban fortalezas y debilidades. En este caso se decidió eliminarlos antes de separar los datos en *train* y *test*. De esta forma se

sabía que las métricas obtenidas en *test* responderían mejor a la *performance* del modelo con datos correctos.

Se partía por tanto de todos los datos. Primero se hizo un análisis con la totalidad de los genes. Los resultados se muestran en el apéndice B.

A continuación se redujo la lista de genes a aquellos que aparecen en el panel de biomarcadores que se utilizó en el artículo y se encuentra en las tablas mmc4 y mmc5 dentro de la carpeta *supplemental information*. Se trata de diez genes.

Se comprobaron si los diez genes aparecían en la base de datos original. Se constató que era así. Sin embargo, uno de esos diez genes fue eliminado en el apartado 4.2.3. Se prosiguió, por tanto, con nueve genes y la totalidad de los pacientes (2351).

A continuación, se ejecutaron varios algoritmos para detección de valores anómalos:

- *T-distributed Stochastic Neighbourhood Embedding* (tSNE): No se observó visualmente separación entre valores normales y valores atípicos (Véase Figura 4.6)
- *Principal component analysis* (PCA): Se eligió visualmente un *threshold* de 2000 y se obtuvo valores anómalos ocho potenciales *outliers* (Véase Figura 4.7)
- *Robust principal component analysis* (rPCA): Se eligió visualmente un *threshold* de 300 y se obtuvo dos valores anómalos potenciales *outliers* (Véase Figura 4.8)
- *Isolation Forest* (IF): Se seleccionaron aquellos valores que tenían una probabilidad mayor del 90 % de ser *outliers*. Se obtuvieron tres puntos.
- *Local Outlier Factor* (LOF): Se seleccionaron aquellos valores que tenían una probabilidad mayor del 90 % de ser *outliers*. Se obtuvo un punto.
- *Density-based spatial clustering of applications with noise* (DBSCAN): Se realizaron diversas pruebas para tratar de encontrar valores de número de vecinos y distancia máxima del vecindario que proporcionaran dos *clusters* principales y una lista de valores atípicos pero sin éxito. Este algoritmo no pareció adecuado para hallar valores anómalos en esta base de datos.

La tabla 4.1 muestra el listado con los valores atípicos encontrados para cada algoritmo.

Valores Atípicos	
PCA	[465, 550, 1206, 1488, 1577, 1768, 1967, 2183]
rPCA	[1752, 1768]
IF	[550, 1699, 1768]
LOF	[1804]

Tabla 4.1: Búsqueda de valores atípicos usando panel de biomarcadores y diferentes algoritmos. Cada número corresponde con el índice dentro del array de observaciones.

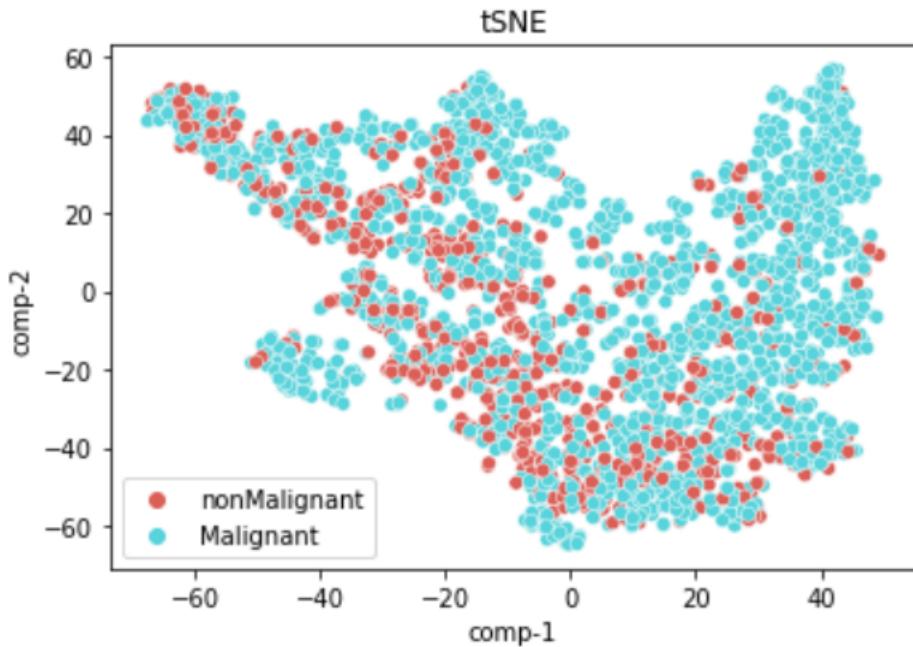


Figura 4.6: tSNE sobre 9 genes

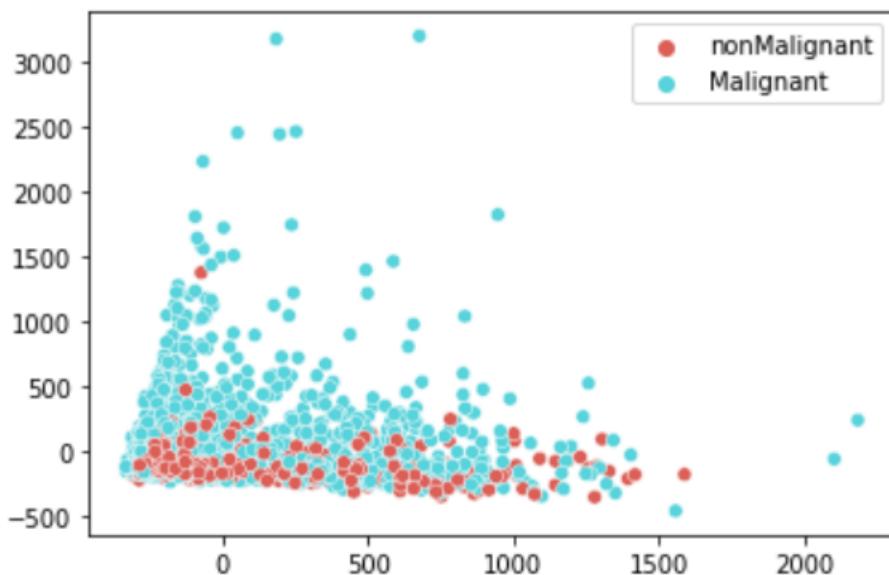


Figura 4.7: PCA sobre 9 genes

Para escoger qué valores anómalos eliminar, se decidió usar como criterio el siguiente: escoger aquellos *outliers* que aparecieran al menos en dos de las listas. Por tanto, quedó la siguiente lista de valores anómalos:

[1768, 550]

Estos pacientes se eliminaron de la base de datos y se pasó al siguiente punto. Como resultado se obtuvo una base de datos en este punto con 2349 pacientes.

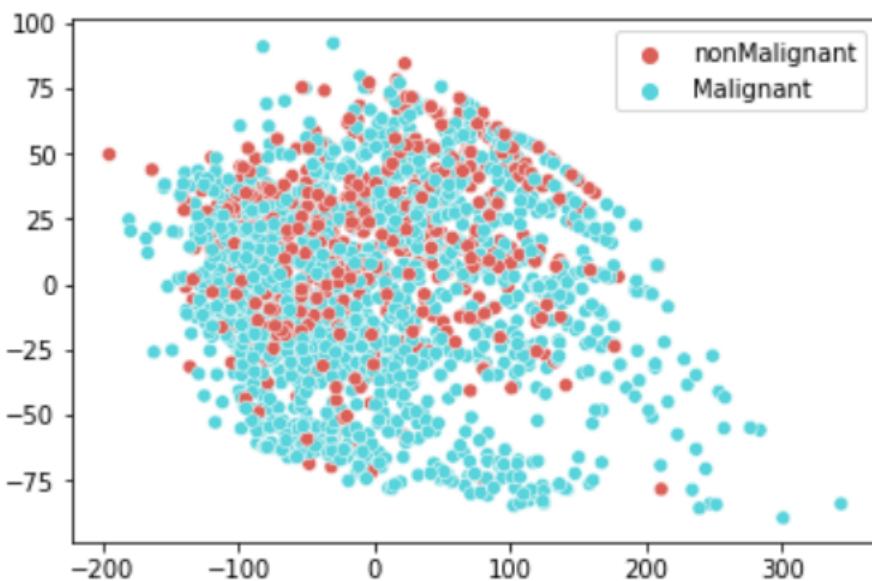


Figura 4.8: rPCA sobre 9 genes

4.5. Partición de datos externa

La partición de datos externa es una técnica utilizada en el aprendizaje automático que consiste en dividir el conjunto de datos en dos partes: una para entrenamiento del modelo y otra para evaluar el rendimiento del modelo en datos no vistos durante el entrenamiento. En este enfoque, los datos de test no están incluidos en la fase de entrenamiento del modelo, lo que permite simular un escenario más realista en el que el modelo se enfrenta a datos desconocidos.

Se denomina partición de datos externa para diferenciarla de la partición de datos interna, que se usa para encontrar los mejores hiperparámetros de cada modelo.

Para la partición de datos externa se decidió usar un *ratio* del 80 % de los datos para entrenamiento y un 20 % para test y el barajado activado. Se obtuvo:

- 1879 pacientes para entrenamiento
- 470 pacientes para test

Se verificó que las clases estaban igualmente balanceadas en la base de datos original y en las particiones, representando la clase *nonMalignant* (pacientes sanos) alrededor de un 30 % en los tres casos.

4.6. Selección de características

Se decidió reducir el número de características para buscar una mejora de la eficiencia computacional del modelo sin sacrificar significativamente la capacidad predictiva. En este caso

se decidió usar métodos estadísticos (en vez de algoritmos del tipo PCA) para mantener la explicabilidad y la portabilidad.

Se decidieron eliminar los siguientes genes:

- Genes cuyos valores nunca superaban un cierto umbral. En este caso se tomó la base de datos de *train*, se calcularon los valores máximos para cada gen y se tomó el valor percentil 25 como umbral. Este valor permitía descartar aquellos genes que tenían una expresión muy baja en la muestra. Esto era importante porque los genes que tienen una expresión muy baja pueden tener un impacto mínimo en el modelo de clasificación y, por lo tanto, no ser relevantes para la detección del cáncer. Se eligió el valor umbral del 25 % como un buen equilibrio entre reducir características y minimizar la pérdida de información. En el apartado de modelado se comprobó con uno de los modelos que esta suposición era correcta.
- Genes cuya varianza no supera un cierto umbral. En este caso se tomó la base de datos de *train*, se calculó la varianza para cada gen y se tomó el valor percentil 25 como umbral. La idea detrás de esta decisión es que los genes que tienen una varianza muy baja no proporcionan información valiosa para el análisis. Se utilizó el umbral del 25 % por ser un umbral usado en la literatura que se esperaba que diera un buen resultado. De nuevo, en el apartado de modelado se comprobó con uno de los modelos que esta suposición era correcta.

Como resultado se obtuvo una base de datos con 2891 genes.

4.7. Modelado

4.7.1. Selección de datos y etiquetas

En este caso se decidió crear modelos utilizando únicamente la información de los genes y descartando el resto de datos: edad, sexo etc.

Además, tal y como se indicó previamente, se realizó un acercamiento al problema de tipo binario (cáncer / sano) en vez de un problema multiclas (18 clases de cáncer). Esto se hizo para simplificar el proceso de modelado, reducir el tiempo de entrenamiento y buscar una mejora de la precisión en la identificación de pacientes con cáncer. Por tanto se utilizó como etiqueta la columna de la clase: *Malignant* y *nonMalignant* y se transformó a valores binarios de la siguiente manera:

Valores de clase	
Malignant	1
nonMalignant	0

Tabla 4.2: Valores de clase.

4.7.2. Selección de métricas

Como se trata de un proceso de clasificación, se optó por utilizar las siguientes métricas:

Exactitud (accuracy): Es el porcentaje de predicciones correctas. Para las clasificaciones binarias, se calcula de la siguiente forma:

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

Donde:

TP: Verdadero Positivo o *True positive*

TN: Verdadero Negativo o *True negative*

FP: Falso Positivo o *False positive*

FN: Falso Negativo o *False negative*

En clasificaciones desbalanceadas (donde hay muchos más ejemplos de una clase que de la otra), este métrica no suele aportar tanta información.

Precisión (precision): Es el porcentaje de positivos predichos que lo son realmente.

Para las clasificaciones binarias, se calcula de la siguiente forma:

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

Sensibilidad (recall o sensitivity): Es el porcentaje de casos positivos que se han predicho de manera satisfactoria. En el caso de predicción de una enfermedad como el cáncer, esta métrica es muy interesante porque ayuda a entender cuántos casos reales de cáncer se consiguen predecir.

Para las clasificaciones binarias, se calcula de la siguiente forma:

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

Especificidad o Tasa negativa verdadera (TNR) (specificity): Es el porcentaje de casos negativos que se han predicho de manera satisfactoria. En el caso de este artículo, era una métrica muy importante para los autores ya que permite asumir que si se tiene una especificidad próxima a 1, el número de falsos positivos será prácticamente cero, y por tanto no se estresará a ningún paciente dando un diagnóstico de enfermedad sin que lo esté. Es decir, si a un paciente obtiene un diagnóstico positivo, es altamente probable que realmente esté enfermo.

Para las clasificaciones binarias, se calcula de la siguiente forma:

$$\text{Especificidad} = \frac{TN}{TN + FP}$$

Curva ROC: es un gráfico que muestra el rendimiento de un modelo de clasificación en todos los umbrales de clasificación. Es decir, es una curva que representa dos parámetros:

- En el eje de abscisas la tasa de falsos positivos ($1 -$ especificidad)

$$\frac{FP}{FP + TN}$$

- En el eje de ordenadas la tasa de verdaderos positivos (sensibilidad)

AUC (Área bajo la curva ROC): mide el área bidimensional completa debajo de la curva ROC completa.

4.7.3. Modelos de Soporte Vectorial

Como se puede leer en [Chang y Lin \(2011\)](#), el *support vector classifier* (SVC) es la versión de modelo SVM para clasificación y trata de encontrar el mejor hiperplano para separar las diferentes clases de manera que la distancia entre los puntos y el hiperplano sea máxima. Véase Figura 4.9.

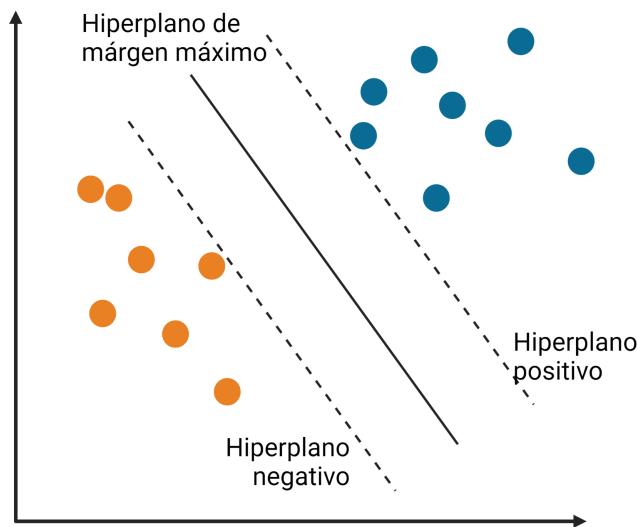


Figura 4.9: Modelo de Soporte Vectorial (creado con BioRender.com)

El modelo de soporte vectorial tiene como ventajas principales su convergencia a mínimo global y la posibilidad de tratar con problemas lineales y no lineales (a través de *kernel trick*). Como inconveniente no es óptimo si el conjunto de datos es muy grande y es menos efectivo si hay clases superpuestas.

El primer paso para modelizar es siempre encontrar los **mejores hiperparámetros** para este modelo. Para ello se utilizó la función *RandomizedSearchCV* que permite indicar una lista de posibles hiperparámetros y realiza un análisis recorriendo algunas de las posibles combinaciones para entregar como resultado el modelo entrenado con la mejor combinación. Esta función realiza una *cross validation* de 5 bolsas.

Los hiperparámetros testeados fueron:

- C: Controla el *trade off* entre tener barreras de decisión suaves y clasificar los puntos correctamente. Es decir indica cuánto de permisivo se es con los errores cometidos durante el entrenamiento. Se probaron los valores [0.1, 1, 10, 100, 1000].
- kernel: Define diferentes formas de hiperplanos, de decir: permite separar datos que no son linealmente separables proyectándolos en una dimensión mayor donde sí lo sean. *Linear* es lineal mientras que *rbf* y *poly* son no lineales.
- gamma: Se utiliza con hiperplanos no lineales y define cuánto exactamente intenta el hiperplano ajustarse a los datos de entrenamiento. Se probaron valores dentro de este rango: [0.1, 1, 10, 100].

Además, se decidió realizar diferentes pruebas cambiando:

- Métricas: Se probó a buscar hiperparámetros maximizando la exactitud o la especificidad. El objetivo era comprobar si, teniendo la especificidad como objetivo a maximizar, se obtenía un valor mayor, y si esto hacía que empeorara (y cuánto) el resto de métricas.
- Datos: Se probó con la base de datos obtenida en el punto 4.7.1. Además se probó a eliminar los pacientes sanos sintomáticos (SC), ya que el artículo plantea que estos pacientes empeoran la *performance* del modelo y se quería comprobar.

Los hiperparámetros optimizados que se encontraron para cada modelo se pueden ver en la tabla 4.3.

Modelo	Métrica	Datos	C	kernel	gamma
1	Accuracy	2881 genes, AC + SC	0.1	poly	1
2	Especificidad	2881 genes, AC + SC	0.1	poly	1
3	Especificidad	2881 genes, AC	0.1	linear	0.1
4	Especificidad (balanced)	2881 genes, AC + SC	1	poly	0.1

Tabla 4.3: Hiperparámetros SVC.

El primer modelo se entrenó usando todo los datos y con el objetivo de maximizar la exactitud. Se creó este modelo como *baseline* de esta prueba aunque se suponía que la exactitud probablemente no era la métrica más adecuada, especialmente porque las clases estaban desbalanceadas. Por esta razón, los otros tres modelos se diseñaron para maximizar la especificidad.

En el caso del segundo modelo, se optimizó los hiperparámetros para maximizar la especificidad. Curiosamente, este modelo entregó los mismos hiperparámetros a pesar de estar ajustado para la métrica de especificidad.

En el tercer modelo se decidió eliminar aquellos pacientes sanos que presentaban síntomas de otras enfermedades que no son cáncer (SC). Se quería validar si mejoraba la especificidad

del modelo, ya que se eliminaba un grupo de muestras que podían tener síntomas similares a los de los pacientes con cáncer (y ciertos perfiles de ARNm más parecidos), pero que no eran relevantes para la clasificación de cáncer en sí misma.

Finalmente, en el cuarto modelo, se decidió usar el parámetro *balanced* para intentar ajustar el hecho de que la base de datos presenta clases desbalanceadas. Al ajustar el parámetro *balanced*, se buscaba equilibrar la importancia de la clase minoritaria (pacientes sanos) con respecto a la clase mayoritaria (pacientes con cáncer), lo cual se pensaba que podía ayudar a mejorar la precisión general de la clasificación.

4.7.4. Modelos de bosque aleatorio (*random forest*)

También se probaron modelos del tipo bosque aleatorio o *random forest* (RF). Como se puede leer en [Breiman \(2001\)](#), *Random Forest* es lo que se llama un método de conjunto (o *ensemble method*), es decir que combina resultados de diferentes árboles de decisión para obtener un mejor resultado final. Esta combinación se puede hacer por ejemplo con la técnica del arbitraje (Véase Figura 4.10).

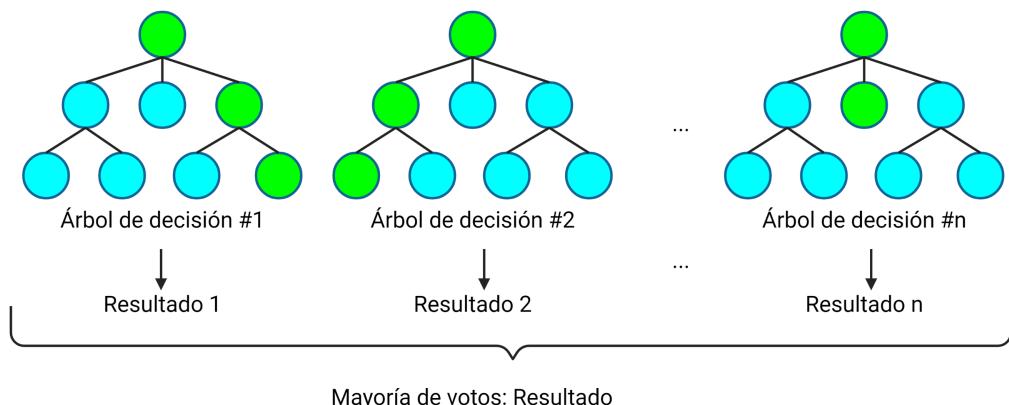


Figura 4.10: Modelo de bosque aleatorio (creado con BioRender.com)

El modelo de bosque aleatorio tiene como ventaja su simplicidad, estabilidad y robustez. Como inconvenientes se tiene un alto coste computacional y menor interpretabilidad respecto a otros modelos.

En este caso también se probaron varios hiperparámetros:

- Número de estimadores (est): es el número de árboles que conformarán el bosque y con los cuales se aplicará posteriormente el arbitraje para realizar la predicción. Se optó por los valores: [200, 400, 600, 800, 1000]
- Máxima profundidad (depth): es la máxima profundidad de cada árbol. En el caso de seleccionar *None*, los nodos se expanden hasta que todas las hojas tienen un único caso. Las opciones con las que se trabajaron en este caso son: [100, 300, 500, *None*].

- *Bootstrap*: Permite trabajar con todas las muestras (False) o solo una porción (True) para construir cada árbol.

De nuevo se realizaron pruebas con diferentes combinaciones de métricas y datos de entrada. En concreto se decidió probar tres modelos: El primero y el segundo se ajustaron buscando maximizar la especificidad, mientras que el tercero, el valor de AUC. Lo que se pretendía era entender si había alguna mejora usando otra métrica que tuviera en cuenta el balance entre clases, no solo la clase negativa. Por otro lado, el primer modelo utilizó la base de pacientes completa, mientras que en el segundo y tercer modelo se quería comprobar el efecto de eliminar a los pacientes sintomáticos (SC).

Los hiperparámetros optimizados encontrados para cada modelo se pueden consultar en la tabla 4.4.

Modelo	Métrica	Datos	n_estimators	max_depth	bootstrap
1	Especificidad	2881 genes, AC + SC	400	300	False
2	Especificidad	2881 genes, AC	400	300	False
3	AUC	2881 genes, AC	600	300	False

Tabla 4.4: Hiperparámetros Random Forest.

4.7.5. Modelos de regresión logística

Se probaron también modelos del tipo regresión logística o *logistic regression* (LOGR). Como se puede leer en [Defazio et al. \(2014\)](#) y en [Fan et al. \(2008\)](#), la regresión logística modela la probabilidad de cada dato de pertenecer a una clase determinada. Por ejemplo, en el caso univariable (Véase Figura 4.11) se podría modelar la clase género en función de la altura, de manera que cuanto más alta es la persona, mayor probabilidad de ser hombre. Normalmente se utiliza un *threshold* de 0.5. La respuesta del modelo sería hombre si el valor de retorno es superior a 0.5 y mujer en caso contrario.

Las principales ventajas de este modelo es su simplicidad, interpretabilidad y rapidez a la hora de entrenar y predecir. El principal inconveniente es precisamente también su simplicidad (como en el caso expuesto) y que presupone independencia entre los diferentes atributos.

Se probaron varios hiperparámetros:

- *solver*: Es el algoritmo que se usa para resolver el problema de optimización. En la documentación se menciona que *lbfgs* es el algoritmo por defecto y que *saga* funciona bien en *datasets* grandes (es más rápido). Por tanto se usaron ambos.
- *penalty*: Especifica la norma de la penalización. Se decidió usar [l2, None] ya que son los que funcionan mejor para los *solver* escogidos, según la documentación.

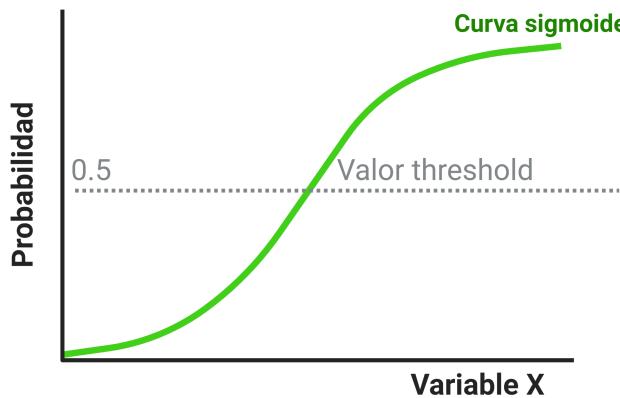


Figura 4.11: Modelo de regresión logística (creado con BioRender.com)

- *class_weight*: permite darle un cierto peso a alguna de las clases. En este caso, como están desbalanceadas (hay más pacientes enfermos que sanos) pareció una buena idea comparar los resultados con datos balanceados (*balanced*) y sin balancear (*None*)
- Número de iteraciones: Interesante comentar que el modelo tuvo problemas de convergencia, y se tuvo que incrementar el número máximo de iteraciones a 10000.

Se realizaron pruebas con diferentes *subsets* de los datos de entrada. En concreto, se testearon los siguientes casos:

En el primero y en el segundo modelo se utilizó el número de genes intermedio que se obtuvo tras aplicar la selección de características (2881). La diferencia entre ellos fue que en el segundo modelo, se eliminó de la lista de pacientes aquellos que eran sanos con síntomas (SC). Como siempre el primer modelo se consideró como la *baseline*.

En el tercer modelo se utilizaron solo los genes provenientes del panel de biomarcadores (8). Se quería comprobar cuánta información se perdía (cuánto se degradaban las métricas) por el hecho de reducir tanto el número de características (genes).

En el cuarto modelo, se utilizó la totalidad de los genes disponibles una vez aplicada la normalización (5251).

Los hiperparámetros optimizados encontrados para cada modelo son como siguen (tabla 4.5):

Modelo	Métrica	Datos	solver	penalty	class_weight
1	Especificidad	2881 genes, AC + SC	saga	l2	balanced
2	Especificidad	2881 genes, AC	saga	l2	balanced
3	Especificidad	8 genes, AC + SC	lbfgs	l2	balanced
4	Especificidad	5251 genes, AC + SC	saga	l2	balanced

Tabla 4.5: Hiperparámetros regresión logística.

4.7.6. Modelos de aprendizaje profundo

Por último se decidió probar modelos del tipo *deep learning*.

Las redes neuronales tienen como ventaja su potencia y su capacidad para modelar las relaciones entre los datos de entrada y salida que no son lineales y que son complejos. Como inconvenientes se tiene un alto coste computacional y menor interpretabilidad respecto los modelos de *machine learning*.

En este caso se probó la arquitectura de red neuronal que se muestra en la Figura 4.12.

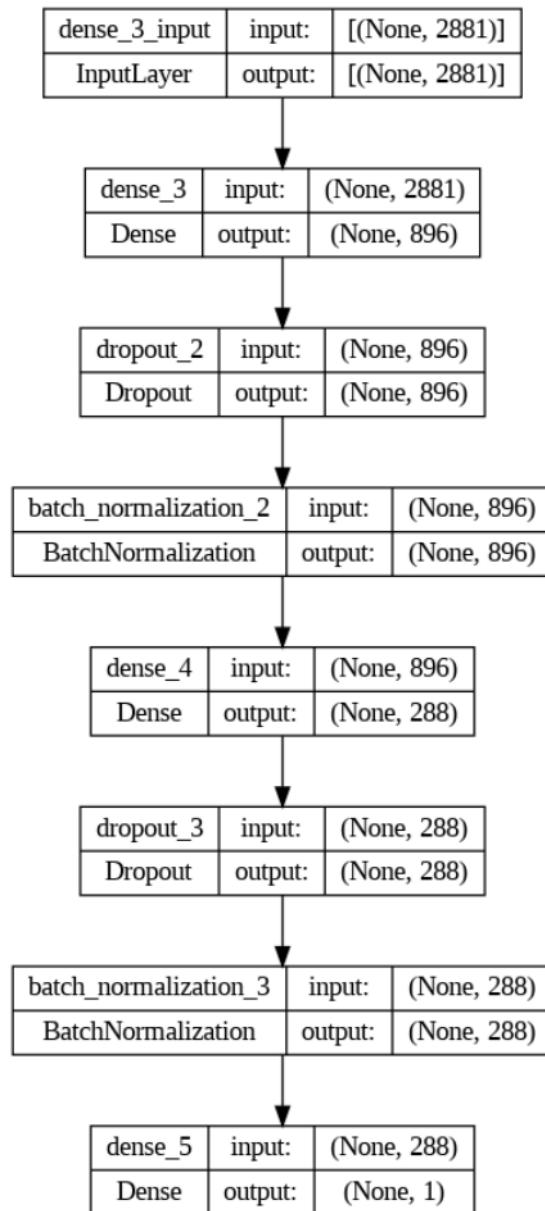


Figura 4.12: Modelo Red Neuronal

En alguna de las pruebas (modelos tres y cuatro), se utilizó la técnica *Synthetic Minority Over-sampling Technique* (SMOTE) que sirve para aumentar el tamaño de los datos de entre-

namiento, especialmente en situaciones en las que hay una desigualdad en la distribución de las clases. Esta técnica se basa en generar nuevas muestras sintéticas para la clase minoritaria a partir de las muestras existentes en esa clase. De esta manera, se aumenta la representación de la clase minoritaria en el conjunto de datos de entrenamiento y se logra un mejor equilibrio entre las clases. También ayuda a prevenir el sobreajuste

Se probaron los siguientes hiperparámetros:

- Número de neuronas de las capas ocultas (N Capa x). En la primera capa oculta, se generó una lista de posibles valores con un rango entre 512 y 2048 con un intervalo de 32. En la segunda capa oculta, se usaron valores entre 64 y 512 con un intervalo de 32. La última capa se definió como una única neurona (clasificación binaria sano/enfermo).
- Porcentaje de *DropOut*. En este caso se decidió optar por un listado de opciones: [0.0, 0.25, 0.5, 0.75].
- Tasa de aprendizaje (*learning rate*, LR) para el algoritmo de cálculo de las pérdidas. En este caso se decidió optar por un listado de opciones: [1e-2, 1e-3, 1e-4].

Para ayudar a encontrar la mejor combinación de hiperparámetros se utilizó el **keras tuner** mediante una hiperbanda. La hiperbanda funciona asignando recursos a configuraciones de hiperparámetros elegidas aleatoriamente y evaluando su desempeño. Luego, descarta la mitad de las configuraciones de peor rendimiento y continúa evaluando las restantes con una cantidad exponencialmente creciente de recursos. Este proceso se repite varias veces, reduciendo así el espacio de búsqueda y acelerando el proceso de ajuste de hiperparámetros. La hiperbanda es más rápida que otras técnicas de optimización como la optimización bayesiana y puede trabajar con hiperparámetros continuos y categóricos.

Además, durante los entrenamientos se utilizaron estas técnicas:

- *Early Stop*: permite parar el entrenamiento si no se ha mejorando la métrica indicada en las últimas x épocas. Con esto se esperaba reducir el tiempo de entrenamiento.
- *Class Weight*: permite dar un mayor peso a la clase minoritaria (en este caso pacientes sanos). Los primeros modelos que se probaron (información no incluida en este trabajo) presentaron alta tendencia a clasificar todos los pacientes como enfermos (especificidad muy baja). Es por ello que se decidió utilizar esta funcionalidad.
- *Checkpointer*: para el modelo final se guardan los pesos de aquellas épocas que dan resultados mejores al mejor resultado encontrado hasta el momento. De esta forma, se eliminaron los resultados obtenidos en las últimas épocas que suelen tener bastante *overfitting*.
- Sobreescribir modelos y limpieza de pesos entre sesiones. Para poder entrenar cada modelo desde un estado aleatorio inicial, y que no estuviera influido por otros resultados previos.

Los hiperparámetros optimizados encontrados para cada modelo se pueden consultar en la tabla 4.6 donde "N Capa x" significa el número de neuronas de la capa x y LR es la tasa de aprendizaje o *learning rate*.

Modelo	Ajuste Hiperparámetros	#N Capa 1	#N capa 2	Drop out 1	Drop out 2	LR
1	Manual	1024	512	0.5	0.5	0.001
2	Keras Tuner	1472	416	0	0.75	0.001
3	Keras Tuner + Data Augmentation	896	288	0	0.75	0.001
4	Keras Tuner + 2x Data Augmentation	608	480	0	0.75	0.001

Tabla 4.6: Hiperparámetros redes neuronales.

Para el caso de aprendizaje profundo se decidió utilizar el número de genes intermedio que se obtuvo tras aplicar la selección de características (2881) y el conjunto de todos los pacientes (una vez eliminados los *outliers*).

El primer modelo se consiguió cambiando los parámetros de manera manual. Posteriormente para el resto de modelos se utilizó el módulo de *keras tuner* para encontrar los mejores hiperparámetros de manera automática. Esto hizo que mejoraran los resultados de validación, pero se seguía teniendo bastante *overfitting*, como se puede apreciar en la Figura 4.13.

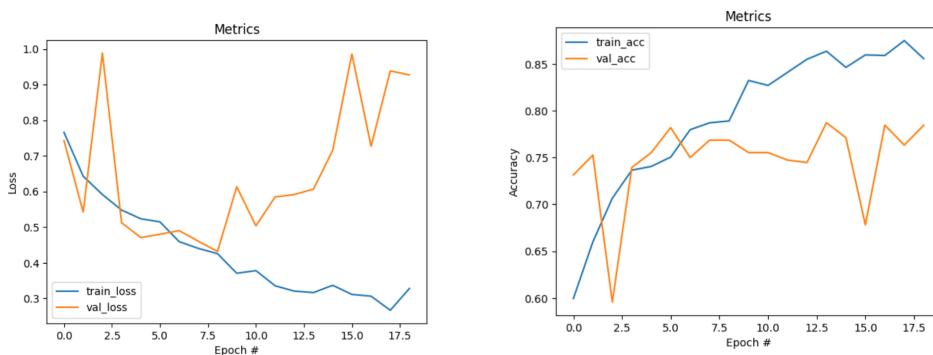


Figura 4.13: Sobreajuste en Redes Neuronales (modelo 2)

Para paliar este punto (además de seleccionar los pesos del mejor modelo tras usar una paciencia de diez), se realizó la misma prueba añadiendo *data augmentation* y se obtuvieron los resultados que se pueden ver en la Figura 4.14 (siguen presentando sobreajuste, aunque algo menos).

Finalmente se probó a añadir más *data augmentation*, pero no pareció ayudar en este caso (véase Figura 4.15):

4.7.7. Entrenamiento modelo definitivo

Cada uno de los modelos mencionados, una vez encontrados los mejores hiperparámetros, fue entrenado con el conjunto de los datos de *train*.

Véase figura 4.16 para un resumen de los pasos realizados.

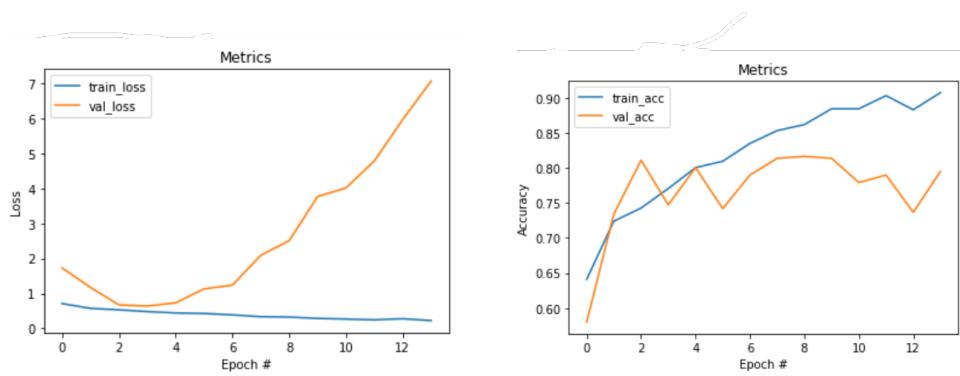


Figura 4.14: Data Augmentation en Redes Neuronales (modelo 3)

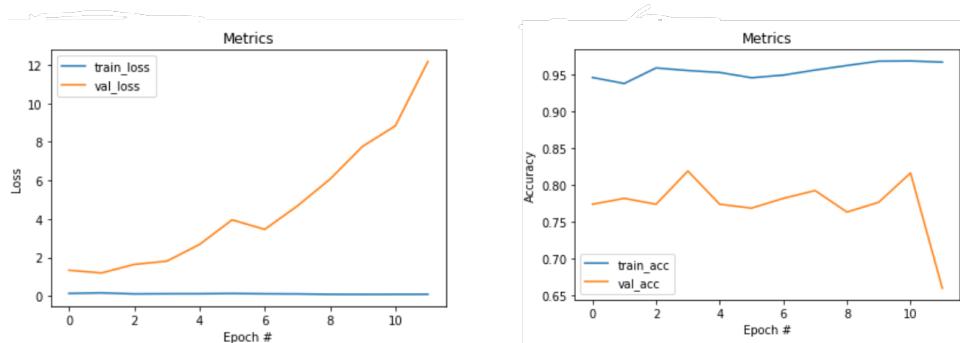


Figura 4.15: Data Augmentation en Redes Neuronales (modelo 4)

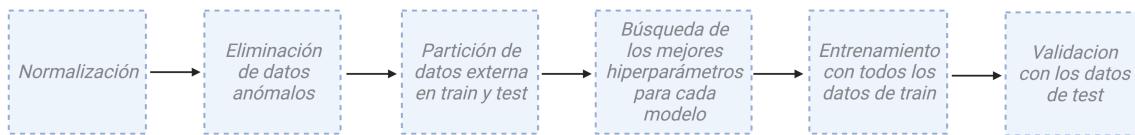


Figura 4.16: Metodología (creado con BioRender.com)

Resultados y Discusión

5

En este apartado, se realizará un análisis detallado de los resultados obtenidos a partir de la predicción y evaluación de cada modelo sobre el *subset* de datos de test. En este análisis se examinarán diversas métricas de evaluación, como la exactitud y la especificidad, para determinar la efectividad y el desempeño del modelo en la tarea de predicción. Finalmente, se llevará a cabo una comparación de los resultados obtenidos en todos los modelos con los resultados que se publican en el artículo [In't Veld et al. \(2022\)](#).

5.1. Resultados

5.1.1. Máquina de soporte vectorial (SVC)

Los resultados que se obtuvieron para cada modelo se pueden ver en la tabla 5.1

Modelo	Exactitud	Especificidad	Sensibilidad	AUC
1	80 %	71 %	85 %	83 %
2	80 %	71 %	85 %	83 %
3	88 %	70 %	93 %	92 %
4	80 %	71 %	85 %	83 %

Tabla 5.1: *Resultados SVC.*

Se puede ver también la gráfica del área bajo la curva ROC del primer modelo (el que obtuvo mejor especificidad) en la Figura 5.1.

Analizando los resultados, se observó lo siguiente:

- Los modelos 1 y 2 entregaron los mismo resultados porque en la fase previa se hallaron los mismo hiperparámetros
- Eliminar los pacientes sanos con síntomas (modelo 3) no mejoró la especificidad aunque sí permitió obtener los mejores resultados en cuanto a exactitud, precisión y AUC (área bajo la curva ROC). El motivo por el que estaba pasando esto (véase fórmula 5.1) es que se había reducido a casi la mitad los verdaderos negativos mientras que se había incrementado ligeramente los verdaderos positivos. Además, se redujeron a la mitad los falsos positivos y negativos.

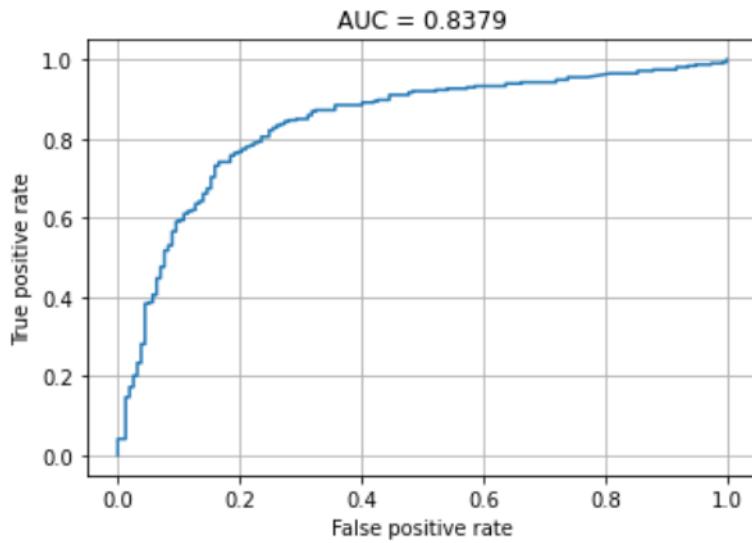


Figura 5.1: AUC SVC (modelo 1)

Matrices de confusión:

$$\text{confusion_matrix} = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}, \text{modelo1} = \begin{bmatrix} 111 & 46 \\ 47 & 266 \end{bmatrix}, \text{modelo3} = \begin{bmatrix} 59 & 25 \\ 23 & 290 \end{bmatrix} \quad (5.1)$$

5.1.2. Bosque aleatorio (*Random forest*)

Los resultados que se obtuvieron para cada modelo se pueden ver en la tabla 5.2.

Modelo	Exactitud	Especificidad	Sensibilidad	AUC
1	76 %	52 %	84 %	84 %
2	82 %	26 %	98 %	88 %
3	82 %	24 %	98 %	88 %

Tabla 5.2: Resultados Random Forest.

Se puede también ver la gráfica del área bajo la curva ROC del primer modelo (el que obtuvo mejor especificidad) en la Figura 5.2.

Analizando los resultados, se observó lo siguiente:

- Bosque aleatorio entregó siempre bajos valores de especificidad.
- Eliminar los pacientes sanos con síntomas (SC) no mejoró la especificidad aunque sí permitió obtener los mejores resultados en cuanto a exactitud, precisión y AUC (área bajo la curva ROC)

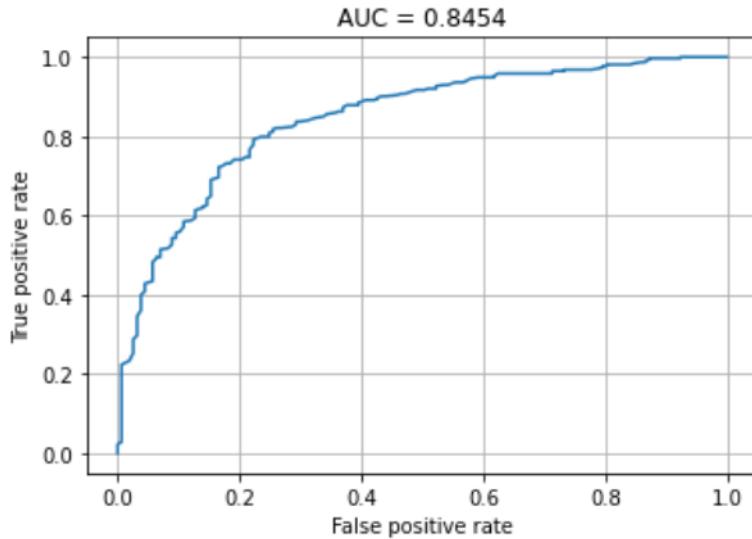


Figura 5.2: **AUC Random Forest** (modelo 1)

5.1.3. Regresión Logística (LOGR)

Los resultados que se obtuvieron de estos modelos se pueden ver en la tabla 5.3

Modelo	Exactitud	Especificidad	Sensibilidad	AUC
1	80 %	85 %	78 %	87 %
2	83 %	82 %	84 %	92 %
3	68 %	80 %	62 %	76 %
4	80 %	85 %	78 %	87 %

Tabla 5.3: **Resultados regresión logística.**

Se puede también ver la gráfica del área bajo la curva ROC del primer modelo (el de mejor especificidad) en la Figura 5.3.

Analizando los resultados, se observó lo siguiente:

- Regresión logística devolvió en todos los casos valores razonablemente buenos de especificidad. Seguramente una de las razones debe ser que se utilizó el hiperparámetro *class_weight*. De hecho en todas las pruebas realizadas, el optimizador escogió siempre *balanced*.
- Eliminar los pacientes sanos con síntomas (SC) no mejoró la especificidad aunque sí permitió obtener los mejores resultados en cuanto a exactitud, precisión y área AUC bajo la curva ROC.
- Reducir el número de genes a 8 arrojó resultados peores en todas las métricas. Esto indicó que el panel de biomarcadores que se menciona en el artículo no era suficiente para expresar todas las diferencias entre pacientes.

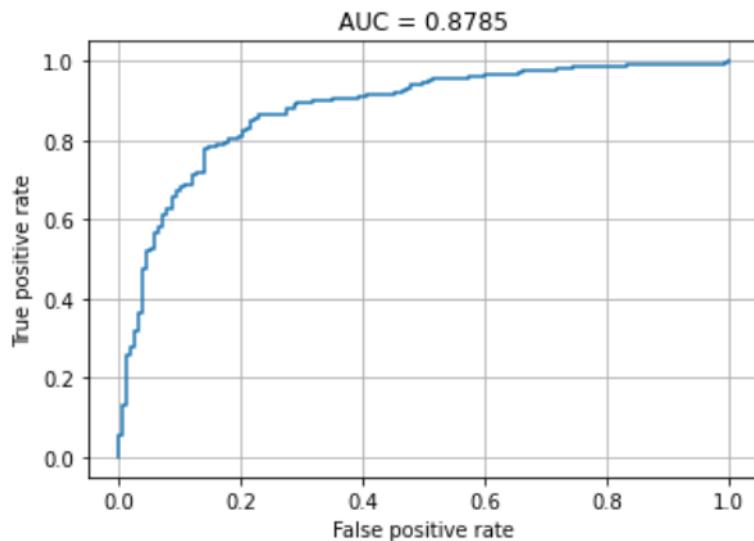


Figura 5.3: AUC regresión logística (modelo 1)

- Utilizar todos los genes (5251) versus solamente un *subset* de 2881 no aportó beneficios en ninguna métrica y en cambio sí que consumió más tiempo de ejecución (ver tabla 5.4). Por tanto, pareció indicar que la reducción de genes realizada en 4.6 no eliminó información relevante.

Modelo	Datos	Tiempo de ejecución por bolsa
1	2881 genes, AC + SC	3-5 mins
2	2881 genes, AC	2-4 mins
3	8 genes, AC + SC	<1 min
4	5251 genes, AC + SC	7-10 mins

Tabla 5.4: Tiempos de ejecución en regresión logística.

5.1.4. Aprendizaje profundo

Los resultados que se obtuvieron se pueden ver en la tabla 5.5.

Modelo	Exactitud	Especificidad	Sensibilidad	AUC
1	76 %	52 %	88 %	83 %
2	80 %	72 %	83 %	86 %
3	74 %	90 %	66 %	85 %
4	80 %	63 %	89 %	85 %

Tabla 5.5: Resultados redes neuronales.

Se añade también la gráfica del área bajo la curva ROC del modelo 3, que corresponde con el de mejor especificidad (véase Figura 5.4).

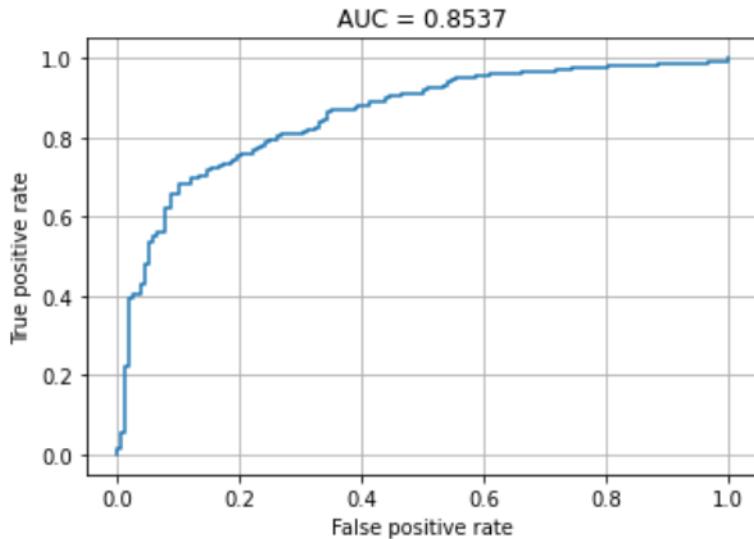


Figura 5.4: AUC Redes Neuronales

Analizando los resultados, se observó lo siguiente:

- Buscar los hiperparámetros de manera manual (modelo 1) fue una tarea muy costosa y entregó los peores resultados.
- Agregar cierta cantidad de datos sintéticos, y especialmente si estos están pensados para balancear las clases (modelo 3), permitió encontrar (por primera vez en este trabajo) un modelo con un nivel de especificidad del 90 %.
- Agregar demasiados datos sintéticos no ayudó al modelo, sino todo lo contrario. Seguramente porque la capacidad de generar datos sintéticos cuando no es fácil deducir la relación entre las características y el *outcome* es limitada.

5.2. Discusión

En la tabla 5.6 se presenta un resumen con los mejores resultados por cada modelo:

Es importante recalcar aquí que los resultados del artículo incluyen la clasificación multiclase mientras que en este trabajo se trata de una clasificación binaria.

Por un lado el mejor modelo de máquina de soporte vectorial (SVC) que se obtuvo, presentó una mejora en exactitud y sensibilidad respecto al artículo, aunque se quedó por detrás en especificidad.

El modelo que peor funcionó fue el de bosque aleatorio.

Por otro lado, el mejor modelo de regresión logística dio buenos resultados comportándose, en las tres métricas, mejor que el modelo compartido en el artículo.

	Exactitud	Especificidad	Sensibilidad
Artículo	46 % - 72 %	99 % (excl. SC) y 78 % (incl. SC)	64 %
SVC	80 %	71 % (incl. SC)	85 %
Random Forest	76 %	52 % (incl. SC)	84 %
LOGR	80 %	85 % (incl. SC)	78 %
Redes Neuronales	74 %	90 % (incl. SC)	66 %

Tabla 5.6: Resumen de resultados.

Por último, las redes neuronales se identificaron como el mejor modelo ya que consiguieron una especificidad mayor que el resto y las tres métricas vencen al modelo del artículo.

Por último, mencionar que en el artículo no se detalla de qué manera han sido capaces de forzar la especificidad hasta el 99 %, más allá de no utilizar los datos de SC. Durante este trabajo, se han probado dos técnicas: búsqueda de hiperparámetros basada en dicha métrica y balanceo de las clases, pero no se ha conseguido llegar a valores tan elevados, incluso eliminando los datos de SC. De hecho, en todos los casos, eliminar de la base de datos los pacientes sanos con síntomas de otras enfermedades que no son cáncer (SC) hace que el modelo funcione mejor en exactitud y AUC, pero no en especificidad.

Además mencionar que reducir el numero de genes a 10 ha hecho que los resultados sean peores.

Conclusiones

6

La detección temprana del cáncer es fundamental para mejorar el tratamiento y probabilidad de superar la enfermedad de los pacientes. Las plaquetas pueden ser una fuente prometedora para la detección del cáncer mediante el análisis del ARN mensajero. En este trabajo se desarrollaron diferentes modelos de aprendizaje supervisado y *deep learning* para predecir la presencia o ausencia de cáncer en pacientes a partir de una biopsia líquida.

Las conclusiones a las que se ha llegado son:

1. Se encontraron modelos con desempeño similar tanto en el ámbito del aprendizaje supervisado (especial mención a la regresión logística) como en el de aprendizaje profundo. Como en aprendizaje supervisado se suele requerir menos coste computacional y se suele contar con mayor explicabilidad, se podría sugerir continuar la investigación siguiendo esa línea.
2. Durante la investigación se pudo constatar que la presencia de desequilibrio en la base de datos afectaba negativamente el rendimiento de los modelos de aprendizaje automático. Dicho impacto se debe a que, en general, estos modelos son diseñados para maximizar la precisión global, lo que puede llevarlos a predecir con mayor frecuencia las clases que están más representadas en la base de datos, produciendo valores de especificidad deficientes.

Ante esta situación, se encontró que la implementación de técnicas específicas para mitigar el efecto del desbalanceo de la base de datos resultó crítica para la obtención de modelos con un mejor comportamiento. De esta manera, se logró mejorar la calidad de las predicciones y, en consecuencia, la efectividad de los modelos.

3. Los resultados obtenidos en este estudio fueron comparables con los del artículo de referencia (si bien el artículo añadía clasificación multiclase), lo que demuestra la validez de los modelos desarrollados.

Limitaciones y Perspectivas de Futuro

7

7.1. Limitaciones

Uno de los principales desafíos en la predicción del cáncer mediante análisis de ARN mensajero de plaquetas ha sido que, a pesar de que se ha utilizado una gran cantidad de datos en este trabajo, la muestra de pacientes podría no ser representativa de la población general. Por lo tanto, los resultados obtenidos podrían no ser aplicables a otros grupos de pacientes o poblaciones (por ejemplo niños o asiáticos). Los modelos de aprendizaje automático estarían entonces sesgados. Es posible que algunos de los resultados obtenidos en este estudio sean específicos para los datos y las características utilizadas.

Por otro lado, este trabajo solo ha tenido en cuenta la clasificación binaria (pacientes enfermos versus sanos). En el estudio original hay información de 18 tipos de cáncer de los que los autores predecían la presencia o ausencia de 16 de ellos y su procedencia (tipo). Podría ser una línea de investigación para futuros modelos predictivos.

Por último, otra limitación es que únicamente se ha utilizado el *dataset* del artículo. Para futuras investigaciones se podría comprobar la validez del modelo con datos provenientes de otras fuentes o estudios.

7.2. Perspectivas de Futuro

Es importante destacar que la detección temprana del cáncer sigue siendo uno de los mayores desafíos en la lucha contra esta enfermedad. Por lo tanto, la investigación en este campo sigue siendo crucial para mejorar las tasas de supervivencia de los pacientes con cáncer y reducir el impacto de esta enfermedad en la sociedad en general.

El uso de la inteligencia artificial en el diagnóstico y tratamiento del cáncer es una de las áreas de investigación más activas en la actualidad. Se espera que en el futuro se puedan utilizar modelos de aprendizaje automático más complejos y sofisticados para mejorar la precisión y la eficacia de la detección temprana y el tratamiento del cáncer. Una posible dirección futura para esta investigación podría ser la exploración de la capacidad de los modelos para detectar la presencia de metástasis en pacientes con cáncer, lo que podría mejorar significativamente la capacidad de los médicos para diseñar tratamientos más efectivos.

El uso de la biopsia líquida, como el análisis de sangre, en la detección del cáncer tiene grandes ventajas sobre los métodos tradicionales invasivos como la biopsia de tejido. Es

probable que en el futuro se desarrollen nuevas técnicas de biopsia líquida que permitan una detección temprana y precisa del cáncer con mayor eficiencia y facilidad.

En resumen, aunque este trabajo ha demostrado que es posible utilizar la información de ARN mensajero en las plaquetas para detectar la presencia de cáncer en pacientes, existen limitaciones importantes a tener en cuenta y se necesitan más investigaciones para mejorar la precisión de los modelos y su capacidad para detectar diferentes tipos de cáncer.

Lista de Acrónimos

ACs controles asintomáticos o *asymptomatic controls*.

ADN ácido desoxirribonucleico.

ARN ácido ribonucleico.

ARNm ARN mensajero.

AUC *Cross-validated area under curve*.

DBSCAN *Density-based spatial clustering of applications with noise*.

FN Falso Negativo o *False negative*.

FP Falso Positivo o *False positive*.

GFF *General Feature Format*.

GRCh37 *Human Gene Annotation - GENCODE Release 19*.

GT *Ground Truth*.

HG19 *human genome 19*.

IF *Isolation Forest*.

Kb Kilobases.

LOF *Local Outlier Factor*.

LOGR regresión logística o *logistic regression*.

PCA *Principal component analysis*.

RF bosque aleatorio o *random forest*.

rPCA *Robust principal component analysis.*

SCs controles sintomáticos o *symptomatic controls*.

SMOTE *Synthetic Minority Over-sampling Technique.*

SVC *support vector classifier.*

SVM *support vector machine.*

TEP plaquetas educadas por tumores o *tumor-educated platelets*.

TN Verdadero Negativo o *True negative*.

TNR Tasa negativa verdadera.

TP Verdadero Positivo o *True positive*.

TPM tránscritos por millón o *transcripts per million*.

tSNE *T-distributed Stochastic Neighbourhood Embedding*.

UCSC University of California Santa Cruz.

UTR región no traducida o *untranslated region*.

Apéndice A

A

En el artículo del que se obtuvo la base de datos, se indica que se ha trabajado con la versión del genoma *human genome 19* (HG19). HG19 es a su vez el alias para *Human Gene Annotation - GENCODE Release 19* (GRCh37) que se puede descargar desde la página web https://www.gencodegenes.org/human/release_19.html indicada en el artículo [Frankish et al. \(2019\)](#). En concreto se descargó la versión en formato GFF.

El formato **GFF** (*General Feature Format*) es un formato de archivo utilizado en genómica y bioinformática para describir las características y anotaciones de los genes y otros elementos genómicos en un genoma secuenciado.

El formato GFF se compone de una serie de líneas de texto, cada una de las cuales describe un elemento genómico diferente, como un gen, una región de un cromosoma, un exón, un intrón, una proteína, una secuencia de ADN, etc. Cada línea de texto en el archivo GFF contiene nueve campos separados por tabuladores, que incluyen información sobre la posición y la identidad del elemento genómico.

Los nueve campos del formato GFF son los siguientes:

- Identificador del cromosoma
- Fuente de la anotación (como el nombre del software o del equipo que realizó la anotación)
- Tipo de característica como: gen, tránscrito, exón, intrón etc.
- Posición de inicio del elemento genómico en el cromosoma
- Posición final del elemento genómico en el cromosoma (ambas incluidas)
- Puntuación de calidad de la anotación (opcional)
- Dirección de la característica (positiva o negativa)
- Fase de codificación de la característica (opcional)
- Atributos adicionales como: el nombre del gen, la fuente de la secuencia, la descripción de la proteína, etc.

Véase las primeras líneas del archivo utilizado en la Figura [A.1](#) (ya convertido a *dataframe* de Pandas). Las columnas tres y cuatro del mencionado archivo se utilizaron para efectuar el

cálculo de las longitudes de los exones, así como para calcular la longitud de los tránscritos correspondientes. Asimismo, la columna ocho fue fragmentada por medio de un delimitador de punto y coma con el propósito de identificar cierta información como por ejemplo el *parent* o progenitor de cada exón y de cada tránscrito, lo cual permitió establecer relaciones de parentesco entre los genes y las isoformas (es decir, los tránscritos).

0	1	2	3	4	5	6	7	8
0	chr1	HAVANA	gene	11869	14412	.	+	. ID=ENSG00000223972.4;gene_id=ENSG00000223972.4...
1	chr1	HAVANA	transcript	11869	14409	.	+	. ID=ENST00000456328.2;Parent=ENSG00000223972.4;...
2	chr1	HAVANA	exon	11869	12227	.	+	. ID=exon:ENST00000456328.2:1;Parent=ENST0000045...
3	chr1	HAVANA	exon	12613	12721	.	+	. ID=exon:ENST00000456328.2:2;Parent=ENST0000045...
4	chr1	HAVANA	exon	13221	14409	.	+	. ID=exon:ENST00000456328.2:3;Parent=ENST0000045...

Figura A.1: Primeras líneas archivo GFF

Apéndice B

B

En este apéndice se presenta el primer acercamiento a la **búsqueda de valores atípicos** en la base de datos. Se cuenta con información de 2351 pacientes y 5251 genes.

B.1. Distribución de los datos

El primer paso es **comprobar la distribución de los datos**. Para ello se seleccionan 3 genes: uno cuya media esté por debajo del 25 % de las medias, otro cuya media esté por encima del 75 % de las medias y un tercero que esté en medio de ambos. A continuación se representan gráficamente sus valores y se observa que tienen largas colas hacia la derecha, lo que es un síntoma claro de no normalidad. Véase Figura B.1.

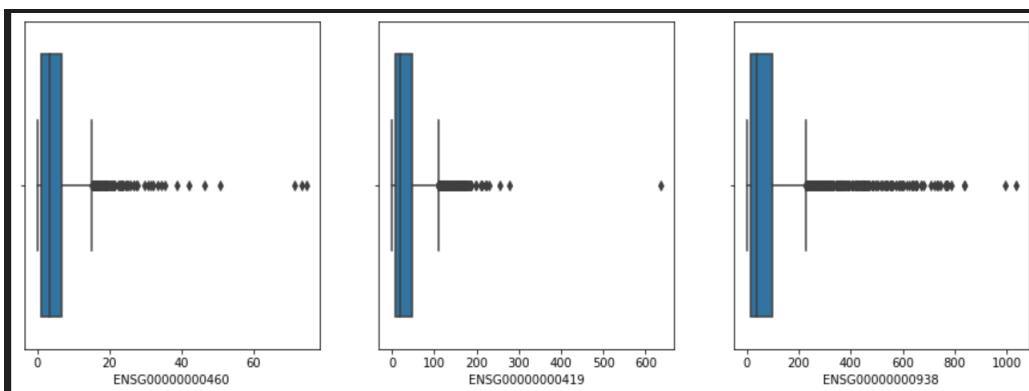


Figura B.1: Distribución de datos de genes

A continuación, se realiza el test **Shapiro-Wilk**, para comprobar (esta vez numéricamente) que los datos no siguen una distribución normal.

B.2. tSNE

Por este motivo, se decide comenzar probando a separar los datos entre *Malignant* y *non-Malignant* aplicando una reducción de dimensionalidad con **T-distributed Stochastic Neighbourhood Embedding (tSNE)** (ya que la PCA es más robusta con datos que siguen una distribución normal). Sin embargo, no se consigue encontrar la forma de separar los grupos y/o de encontrar pacientes anómalos. Véase Figura B.2.

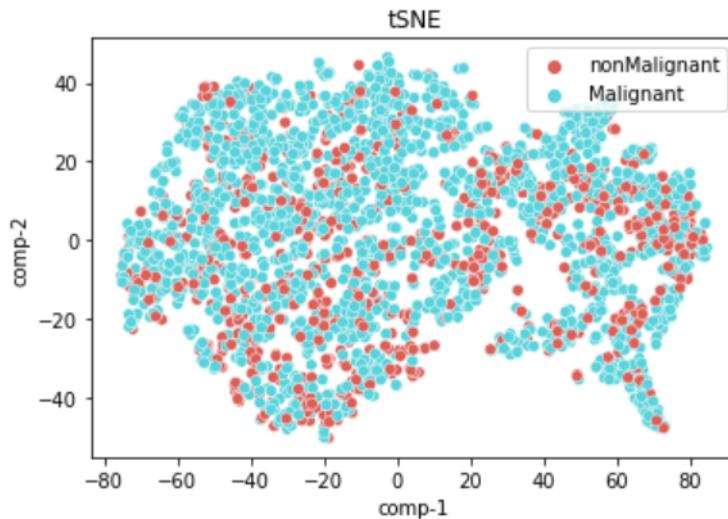


Figura B.2: Algoritmo no supervisado t-SNE aplicado a la búsqueda de valores anómalos

B.3. PCA

A continuación se aplica una reducción de dimensionalidad con *Principal component analysis* (PCA). En este caso sí se aprecia visualmente un par de puntos que parece estar alejado del resto. Véase Figura B.3. Se toma nota de sus valores.

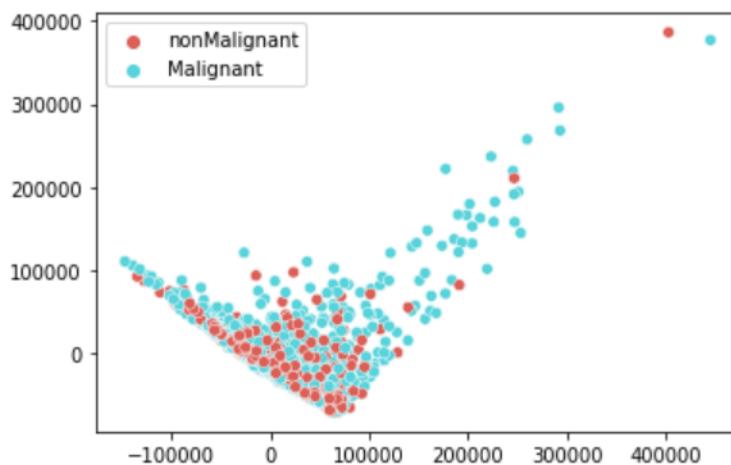


Figura B.3: Algoritmo no supervisado PCA aplicado a la búsqueda de valores anómalos

B.4. rPCA

En el artículo [Chen et al. \(2020\)](#) se introduce una problemática en el uso de ARN secuenciando: la introducción (involuntaria e indeseada) de errores aleatorios o sistemáticos en alguno de los procesos necesarios para realizar esta secuenciación (ver [Best et al. \(2019\)](#) para entender más sobre estos procesos). Esto lleva a tener muestras anómalas que perjudican el modelo.

El artículo también menciona las precauciones que deben tomarse a la hora de eliminar valores atípicos, ya que las anomalías que provengan de diferencias biológicas no deberían ser eliminadas. Únicamente aquellas que provengan de fallos técnicos.

El artículo menciona una alternativa a la inspección visual de un *biplot* de PCA: usar *Robust principal component analysis* (rPCA). Con **rPCA** se pretende encontrar componentes principales que no estén influidas por los *outliers* para posteriormente encontrar estos puntos anómalos. Para este trabajo se ha utilizado el código procedente de este repositorio de *GitHub*: <https://github.com/dganguli/robust-pca>. Una vez aplicado el algoritmo, se encuentran dos *outliers* que coinciden con los vistos en la PCA convencional. Véase Figura B.4. Se toma nota de sus valores.

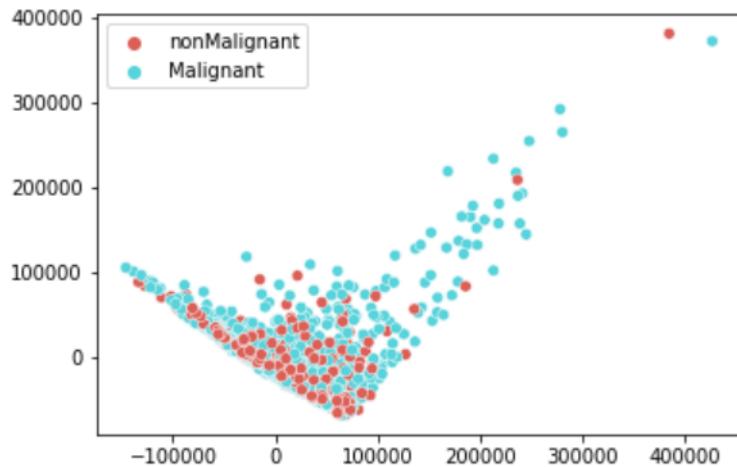


Figura B.4: Algoritmo no supervisado Robust PCA aplicado a la búsqueda de valores anómalos

B.5. *Isolation Forest*

A continuación, tras la lectura del artículo Zhao et al. (2019), se pasa a utilizar la librería PyOD de Python, enfocada a la detección de valores anómalos en datos multivariante.

El primer método que se emplea es el *Isolation Forest*. Es un método no supervisado para identificar valores atípicos. Funciona para cualquier base de datos, pero está especialmente diseñado para datos con alto número de dimensiones y complejidad, como es este caso. Se seleccionan aquellos valores que tienen una probabilidad superior al 90 % de ser *outlier*. Se obtienen dos valores. Se toma nota de los pacientes.

B.6. *Local Outlier Factor*

Se repite el proceso anterior pero esta vez con el algoritmo *Local Outlier Factor*. Está especialmente pensado para bases de datos donde sus puntos suelen agruparse en *clusters*. También funciona normalmente bien con datos con alta dimensionalidad. En este caso se obtienen 3 resultados. Se toma nota de los pacientes.

B.7. DBSCAN

Por último, se intenta ajustar el algoritmo DBSCAN para minimizar las diferencias entre *ground truth* y los *clústeres* arrojados por el algoritmo. Este algoritmo cuenta con dos variables a ajustar. Por un lado la distancia mínima para considerar que dos muestras son vecinas. Por otro el número mínimo de vecinos de una muestra para ser considerada punto nuclear.

En este caso, a pesar de hacer bastantes pruebas, no se consigue realizar un *clustering* en dos grandes grupos (lo cual tiene sentido, ya que se sabe que no se trata de un problema fácilmente resoluble). La estrategia que se sigue en este caso, por tanto, es buscar un *cluster* de muestras y marcar como anómalas aquellas que no se parecen al resto. Con esta técnica se aíslan 8 valores. Se toma nota.

B.8. Resultados

La tabla B.1 muestra el listado con los valores atípicos encontrados para cada algoritmo.

Valores Atípicos	
PCA	[1671, 2100]
rPCA	[1671, 2100]
IF	[126, 313]
LOF	[57, 1963, 1967]
DBSCAN	[57, 812, 819, 1671, 1961, 1963, 1967, 2100]

Tabla B.1: Búsqueda de valores atípicos usando todos los genes y diferentes algoritmos.
Cada número corresponde con el índice dentro del array de observaciones.

Se decide usar como criterio el siguiente: escoger aquellos *outliers* que aparezcan al menos en dos de las listas. Por tanto, quedaría la siguiente lista de valores anómalos:

[1671, 1963, 1967, 2100, 57]

Bibliografía

- Best, M. G., In't Veld, S. G., Sol, N., y Wurdinger, T. (2019). Rna sequencing and swarm intelligence-enhanced classification algorithm development for blood-based disease diagnostics using spliced blood platelet rna. *Nature protocols*, 14(4):1206–1234.
- Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.
- Chang, C.-C. y Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.
- Chen, X., Zhang, B., Wang, T., Bonni, A., y Zhao, G. (2020). Robust principal component analysis for accurate outlier sample detection in rna-seq data. *Bmc Bioinformatics*, 21(1):1–20.
- Copelli, S. B. (2010). *Genética: Desde la herencia a la manipulación de los genes*. Fundación de Historia Natural Félix de Azara, 1 edition.
- Defazio, A., Bach, F., y Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27.
- Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., y Lin, C.-J. (2008). Liblinear: A library for large linear classification. *the Journal of machine Learning research*, 9:1871–1874.
- Frankish, A., Diekhans, M., Ferreira, A.-M., Johnson, R., Jungreis, I., Loveland, J., Mudge, J. M., Sisu, C., Wright, J., Armstrong, J., et al. (2019). Gencode reference annotation for the human and mouse genomes. *Nucleic acids research*, 47(D1):D766–D773.
- In't Veld, S. G., Arkani, M., Post, E., Antunes-Ferreira, M., D'Ambrosi, S., Vessies, D. C., Vermunt, L., Vancura, A., Muller, M., Niemeijer, A.-L. N., et al. (2022). Detection and localization of early-and late-stage cancers using platelet rna. *Cancer cell*, 40(9):999–1009.
- Liu, T., Wang, X., Guo, W., Shao, F., Li, Z., Zhou, Y., Zhao, Z., Xue, L., Feng, X., Li, Y., et al. (2022). Rna sequencing of tumor-educated platelets reveals a three-gene diagnostic signature in esophageal squamous cell carcinoma. *Frontiers in Oncology*, page 1727.
- Łukasiewicz, M., Pastuszak, K., Łapińska-Szumczyk, S., Różański, R., Veld, S. G. I., Bieńkowski, M., Stokowy, T., Ratajska, M., Best, M. G., Würdinger, T., et al. (2021). Diagnostic accuracy of liquid biopsy in endometrial cancer. *Cancers*, 13(22):5731.
- Rodríguez Arnaiz, R., Castañeda Sortibrán, A., y Ordáz Téllez, M. G. (2016). *Conceptos Básicos de Genética*. UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, 1 edition.
- Varkey, J. y Nicolaides, T. (2021). Tumor-educated platelets: A review of current and potential applications in solid tumors. *curaeus*, 13 (11), e19189.
- Xiao, R., Liu, C., Zhang, B., y Ma, L. (2022). Tumor-educated platelets as a promising biomarker for blood-based detection of renal cell carcinoma. *Frontiers in Oncology*, page 689.
- Zhao, Y., Nasrullah, Z., y Li, Z. (2019). Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7.